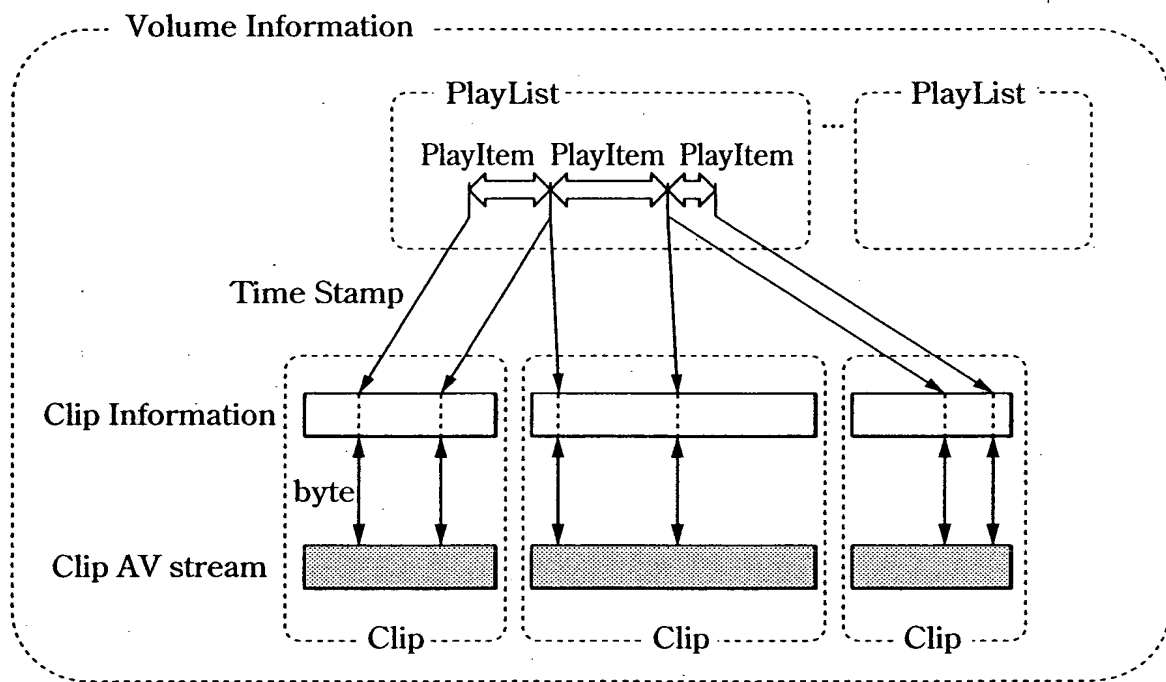
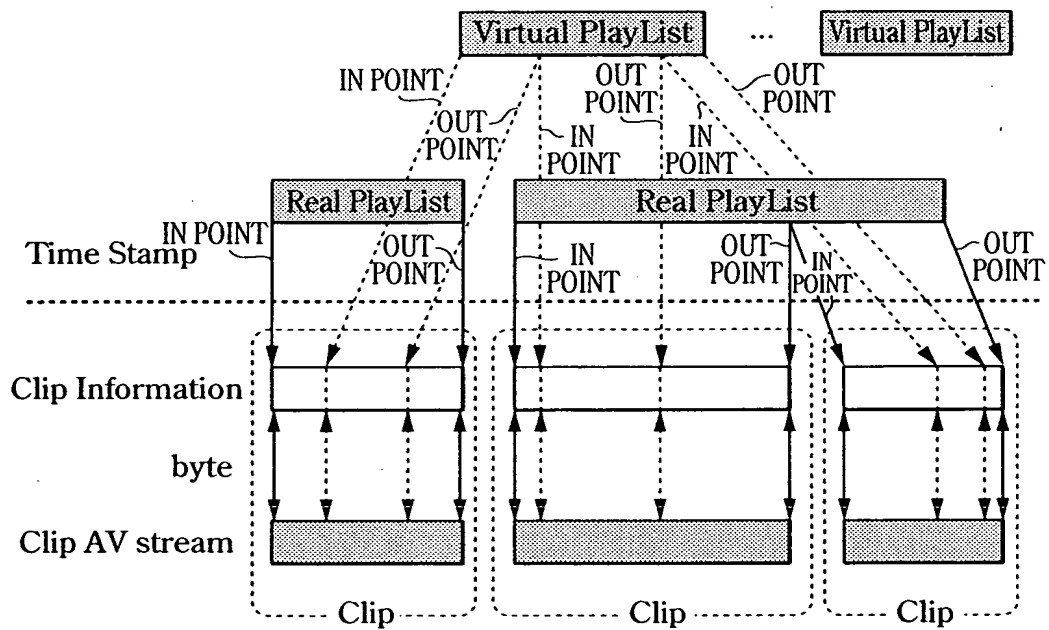
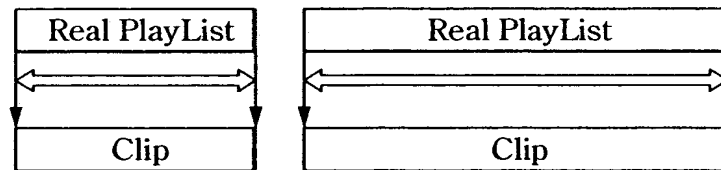


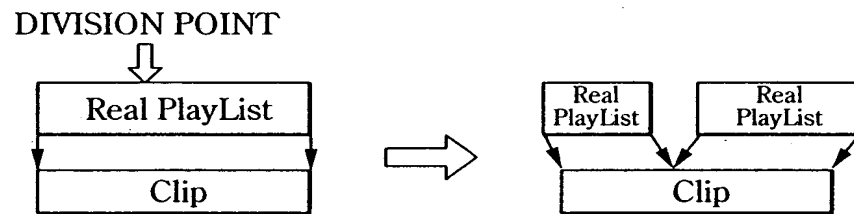
FIG.1

**FIG.2**

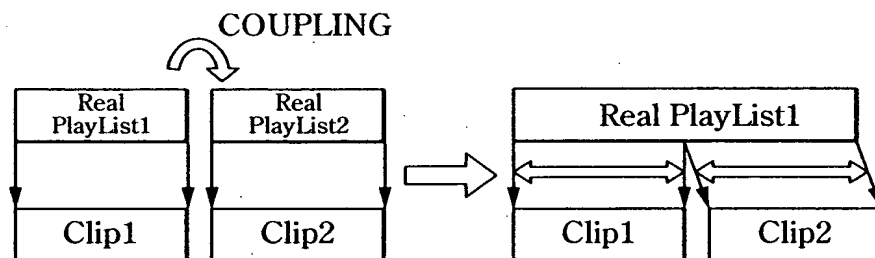
**FIG.3**



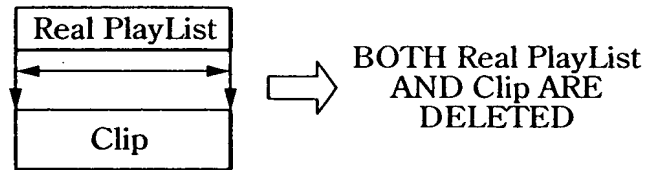
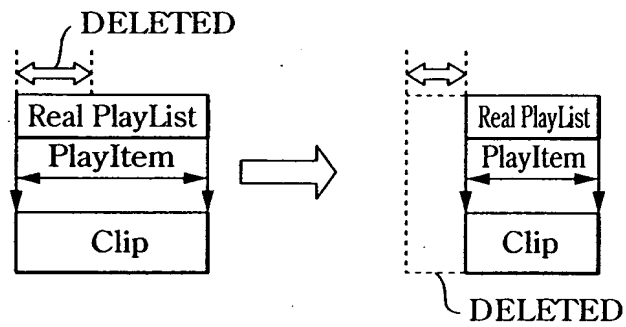
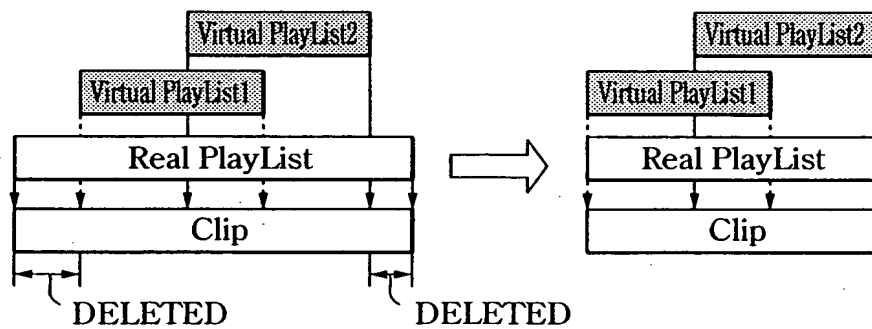
**FIG.4A**

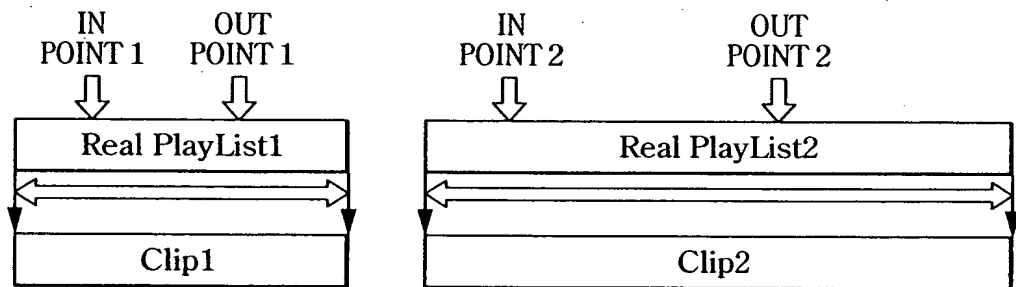
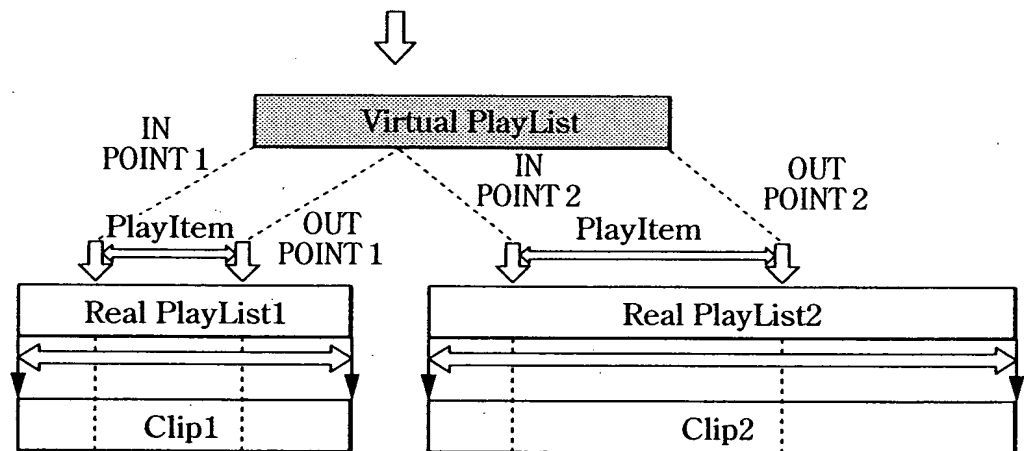


**FIG.4B**



**FIG.4C**

**FIG.5A****FIG.5B****FIG.5C**

**FIG.6A****FIG.6B**

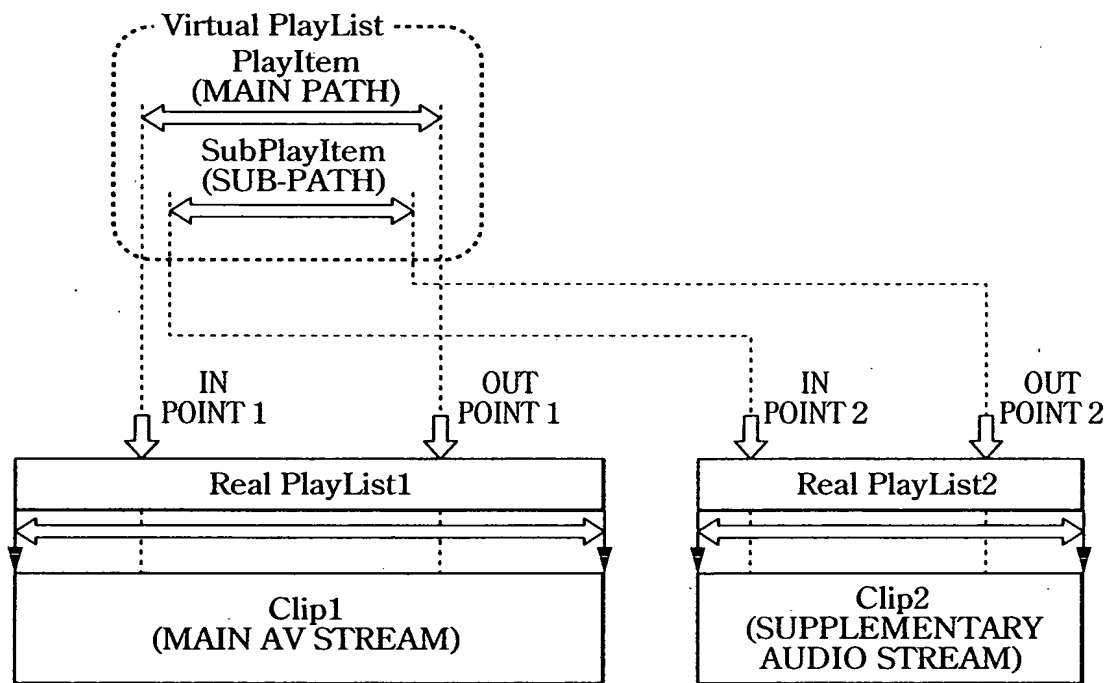
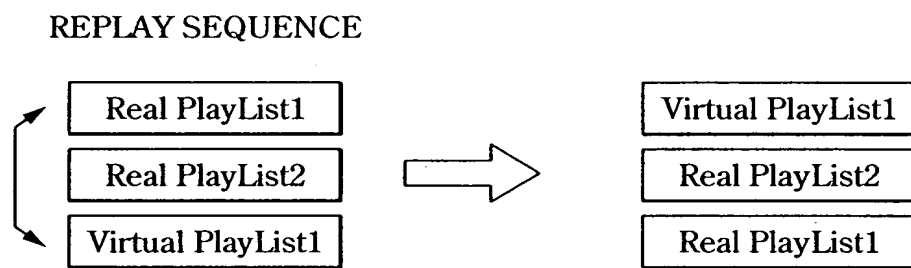
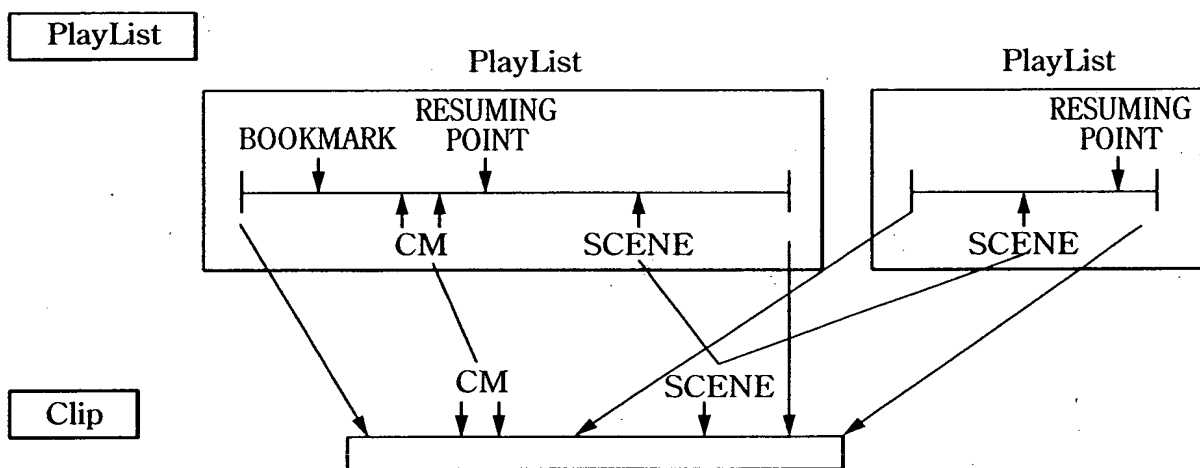


FIG.7

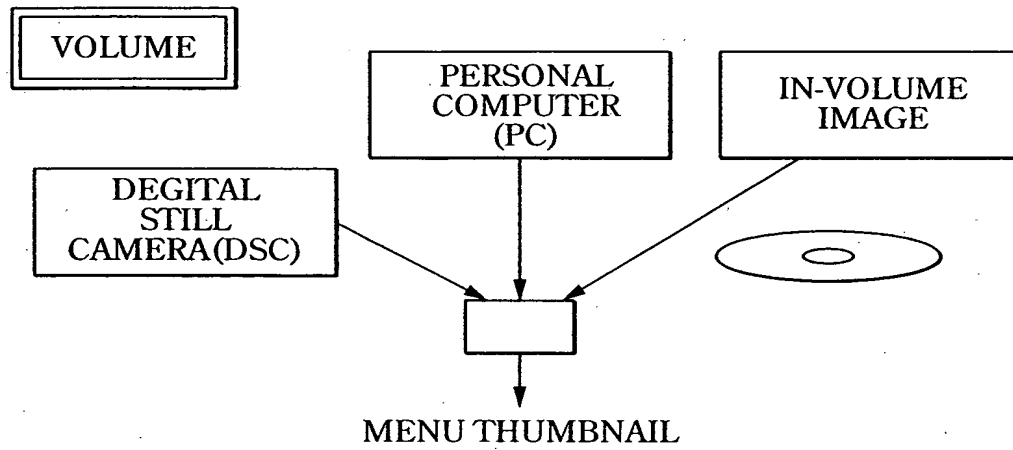


**FIG.8**

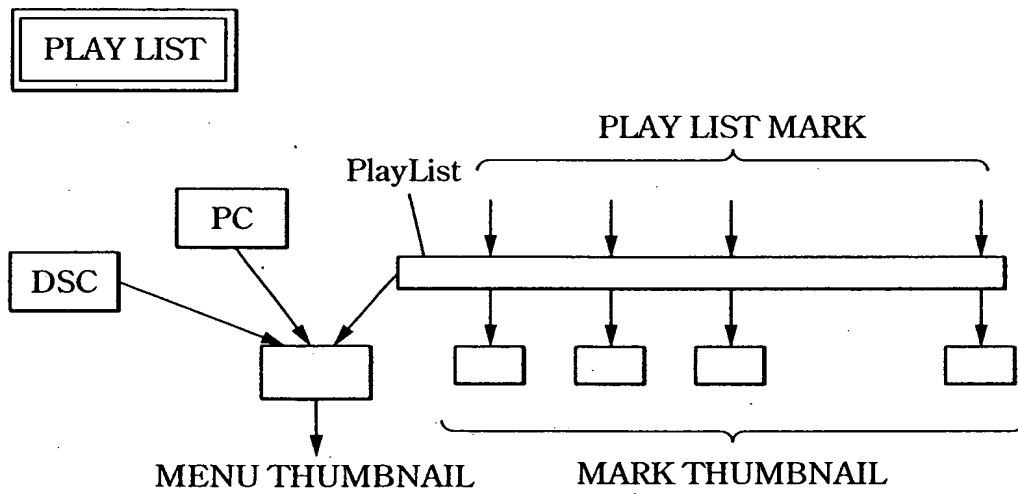




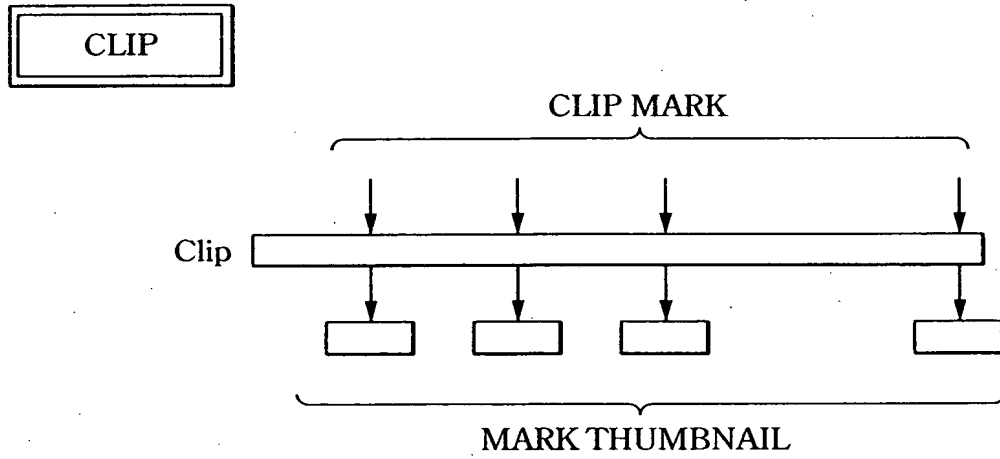
**FIG.9**



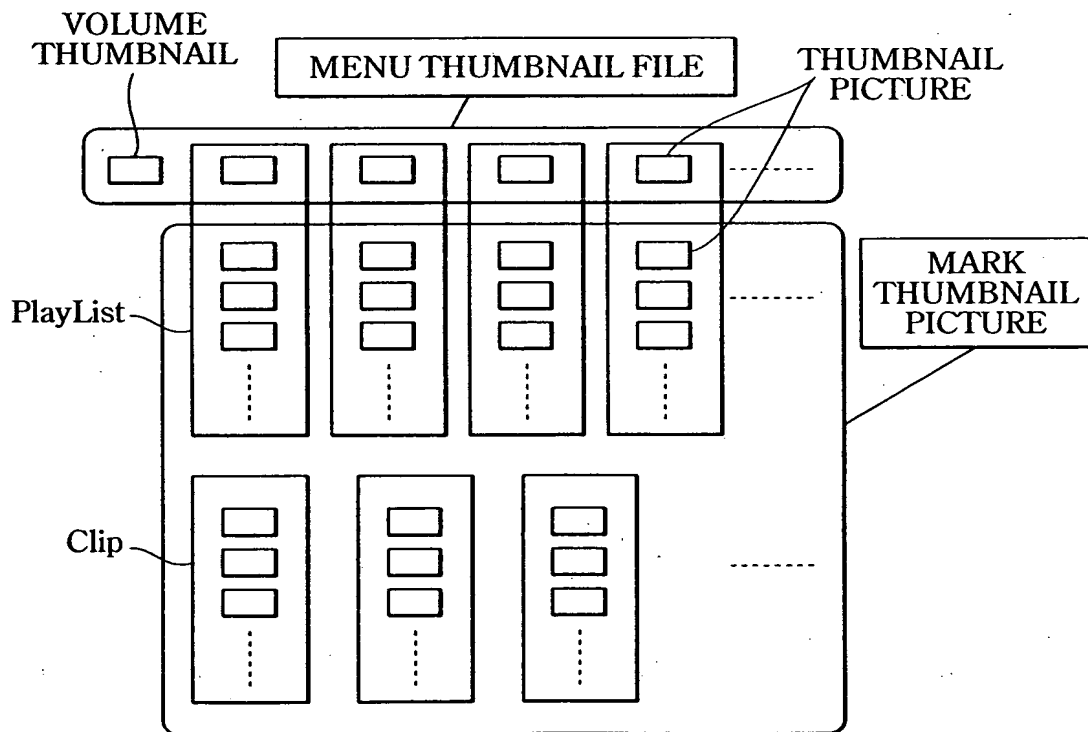
**FIG.10**



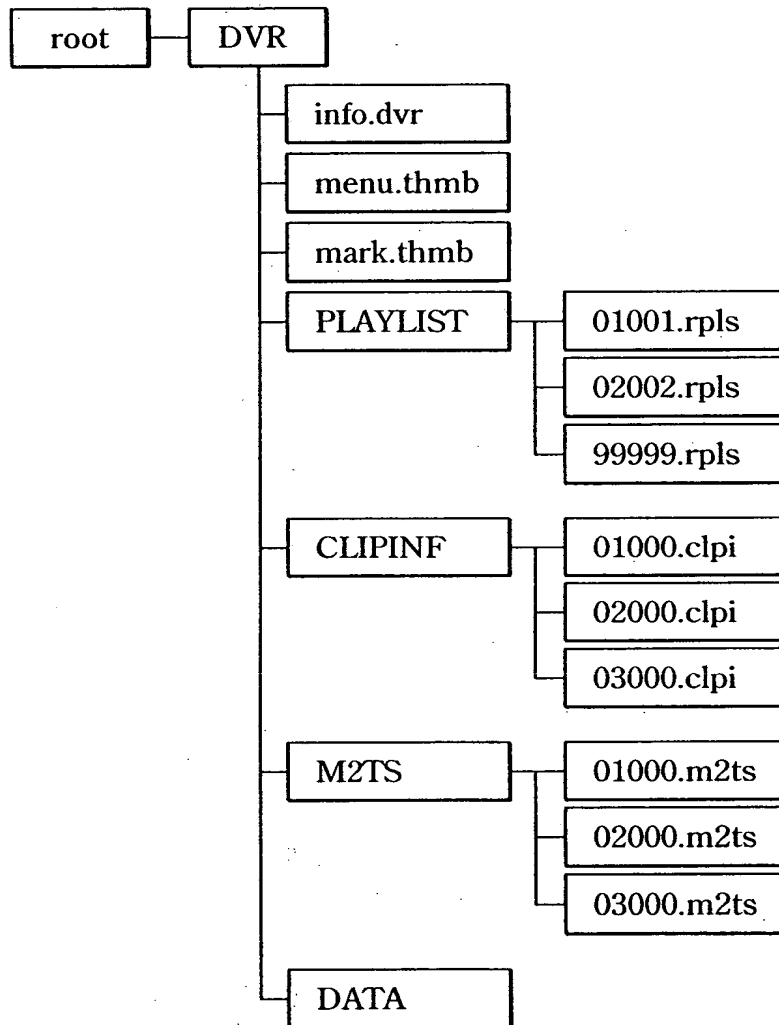
**FIG.11**



**FIG.12**



**FIG.13**

**FIG.14**

SYNTAX	NUMBER OF BYTES	ABBREVIATION
<b>info.dvr {</b>		
<b>TableOfPlayLists_Start_address</b>	32	uimsbf
<b>MakersPrivateData_Start_address</b>	32	uimsbf
reserved	192	bslbf
<b>DVRVolume()</b>		
for (i=0;i<N1;i++){		
padding_word	16	bslbf
}		
<b>TableOfPlayLists()</b>		
for (i=0;i<N2;i++){		
padding_word	16	bslbf
}		
<b>MakersPrivateData()</b>		
<b>}</b>		

FIG.15

SYNTAX	NUMBER OF BYTES	ABBREVIATION
DVRVolume(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>ResumeVolume()</b>		
<b>UIAppInfoVolume()</b>		
}		

FIG.16

SYNTAX	NUMBER OF BYTES	ABBREVIATION
ResumeVolume0{		
reserved	15	bslbf
valid_flag	1	bslbf
resume_PlayList_name	8*10	bslbf
}		

FIG.17

SYNTAX	NUMBER OF BYTES	ABBREVIATION
UIAppInfoVolume0{		
<b>character_set</b>	8	bslbf
<b>name_length</b>	8	uimsbf
<b>Volume_name</b>	8*256	bslbf
<b>reserved</b>	15	bslbf
<b>Volume_protect_flag</b>	1	bslbf
<b>PIN</b>	8*4	bslbf
<b>ref_thumbnail_index</b>	16	uimsbf
<b>reserved_for_future_use</b>	256	bslbf
}		

FIG.18



VALUE	CHARACTER LETTER ENCODING
0x00	Reserved
0x01	ISO/IEC 646 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

**FIG.19**

SYNTAX	NUMBER OF BYTES	ABBREVIATION
TableOfPlayLists(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_PlayLists</b>	16	uimsbf
for (i=0; i< <i>number_of_PlayLists</i> ; i++){		
<b>PlayList_file_name</b>	8*10	bslbf
}		
}		

FIG.20

SYNTAX	NUMBER OF BYTES	ABBREVIATION
TableOfPlayLists(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_PlayLists</b>	16	uimsbf
for (i=0; i< <i>number_of_PlayLists</i> ; i++){		
<b>PlayList_file_name</b>	8*10	bslbf
<b>UIAppInfoPlayList()</b>		
}		
}		

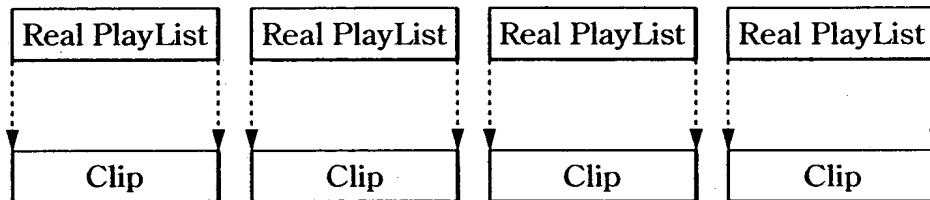
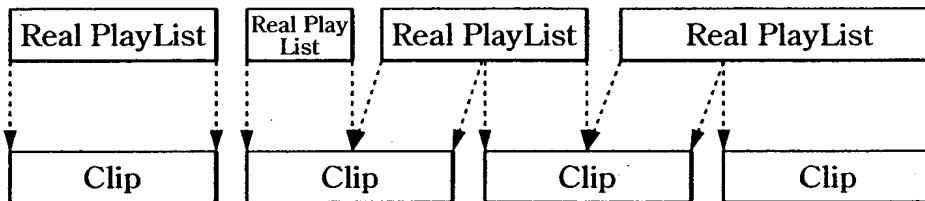
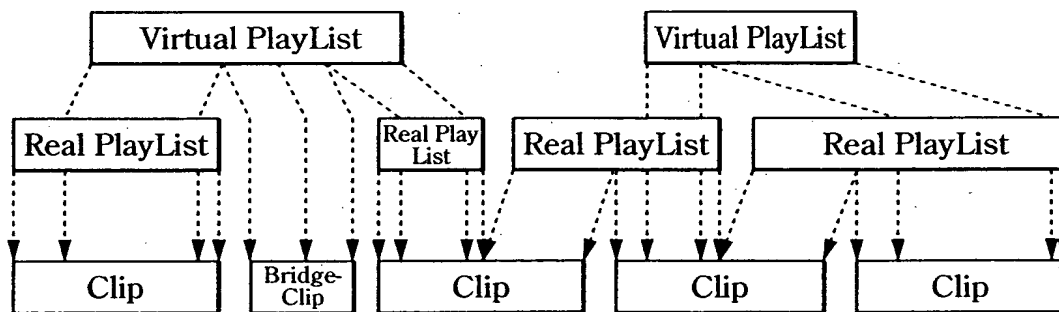
FIG.21

SYNTAX	NUMBER OF BYTES	ABBREVIATION
MakersPrivateData(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
if (length !=0){		
<b>mpd_blocks_start_address</b>	32	uimsbf
<b>number_of_maker_entries</b>	16	uimsbf
<b>mpd_block_size</b>	16	uimsbf
<b>number_of_mpd_blocks</b>	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_maker_entries; i++){		
<b>maker_ID</b>	16	uimsbf
<b>maker_model_code</b>	16	uimsbf
<b>start_mpd_block_number</b>	16	uimsbf
reserved	16	bslbf
<b>mpd_length</b>	32	uimsbf
}		
<b>stuffing_bytes</b>	8*2*L1	bslbf
for(j=0; j<number_of_mpd_blocks; j++){		
<b>mpd_block</b>	mpd_block_size*1024*8	
}		
}		
}		

FIG.22

SYNTAX	NUMBER OF BYTES	ABBREVIATION
xxxxxx.rpls / yyyyyy.vpls {		
<b>PlayListMark_Start_address</b>	32	uimsbf
<b>MakersPrivateData_Start_address</b>	32	uimsbf
reserved	192	bslbf
<b>PlayList()</b>		
for (i=0;i<N1;i++){		
<b>padding_word</b>	16	bslbf
}		
<b>PlayListMark()</b>		
for (i=0;i<N2;i++){		
<b>padding_word</b>	16	bslbf
}		
<b>MakersPrivateData()</b>		
}		

FIG.23

**FIG.24A****FIG.24B****FIG.24C**

SYNTAX	NUMBER OF BYTES	ABBREVIATION
PlayList(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>PlayList_type</b>	8	uimsbf
<b>CPI_type</b>	1	bslbf
reserved	7	bslbf
<b>UIAppInfoPlayList()</b>		
<b>number_of_PlayItems</b> // main path	16	uimsbf
if (<Virtual PlayList>){		
<b>number_of_SubPlayItems</b> // sub path	16	uimsbf
}else{		
reserved	16	bslbf
}		
for (PlayItem_id=0; PlayItem_id<number_of_PlayItems; PlayItem_id++){		
<b>PlayItem()</b> //main path		
}		
if (<Virtual PlayList>){		
if (CPI_type==0 && PlayList_type==0){		
for (i=0; i<number_of_SubPlayItems; i++)		
<b>SubPlayItem()</b> //sub path		
}		
}		
}		

FIG.25

PlayList_type	MEANING
0	PLAY LIST FOR AV RECORDING ALL CLIPS REFERENCED IN THIS PLAY LIST MUST CONTAIN ONE OR MORE VIDEO STREAMS
1	PLAY LIST FOR AUDIO RECORDING ALL CLIPS REFERENCED IN THIS PLAYLIST MUST CONTAIN ONE OR MORE AUDIO STREAMS AND MUST NOT CONTAIN VIDEO STREAMS
2-255	reserved

**FIG.26**



SYNTAX	NUMBER OF BYTES	ABBREVIATION
UIAppInfoPlayList2(){		
<b>character_set</b>	8	bslbf
<b>name_length</b>	8	uimsbf
<b>PlayList_name</b>	8*256	bslbf
reserved	8	bslbf
<b>record_time_and_date</b>	4*14	bslbf
reserved	8	bslbf
<b>duration</b>	4*6	bslbf
<b>valid_period</b>	4*8	bslbf
<b>maker_id</b>	16	uimsbf
<b>maker_code</b>	16	uimsbf
reserved	11	bslbf
<b>playback_control_flag</b>	1	bslbf
<b>write_protect_flag</b>	1	bslbf
<b>is_played_flag</b>	1	bslbf
<b>archive</b>	2	bslbf
<b>ref_thumbnail_index</b>	16	uimsbf
<b>reserved_for_future_use</b>	256	bslbf
}		

FIG.27

write_protect_flag	MEANING
0b	THE PlayList CAN BE ERASED FREELY
1b	THE PlayList CONTENTS SHOULD NOT BE ERASED NOR CHANGED EXCEPT write-protect-flag

**FIG.28A**

is_played_flag	MEANING
0b	THE PlayList HAS NOT BEEN REPRODUCED SINCE ITS RECORDING
1b	THE PlayList WAS ONCE REPRODUCED SINCE ITS RECORDING

**FIG.28B**

archive	MEANING
00b	NO MEANING DEFINED
01b	ORIGINAL
10b	COPY
11b	reserved

**FIG.28C**

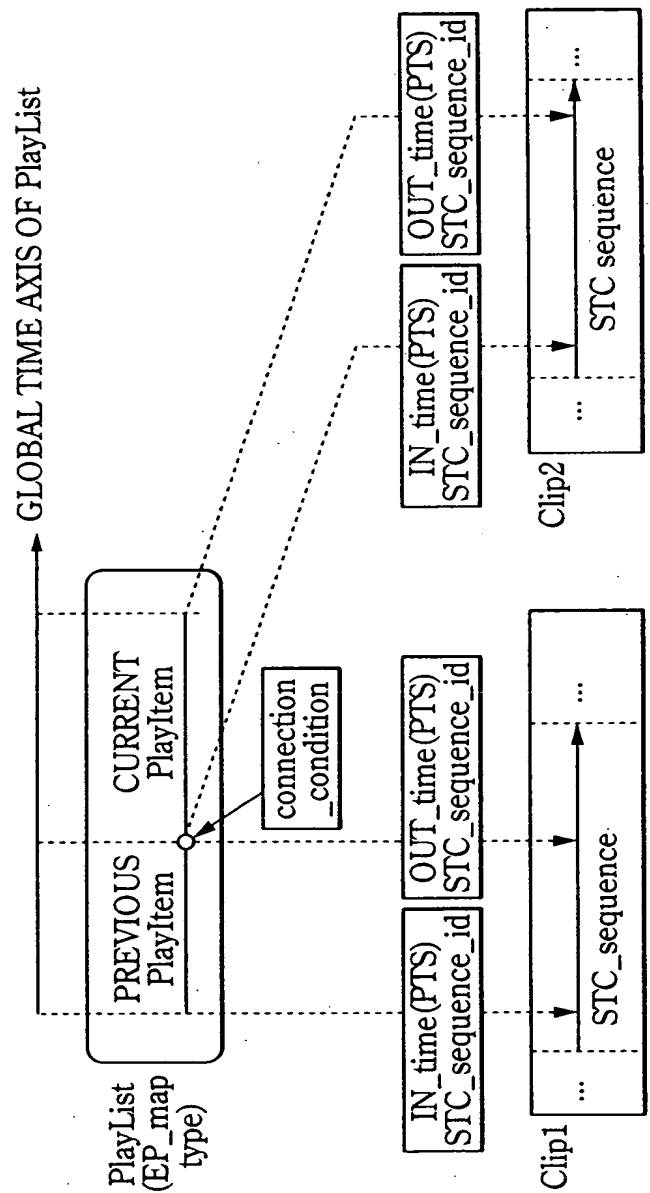


FIG.29

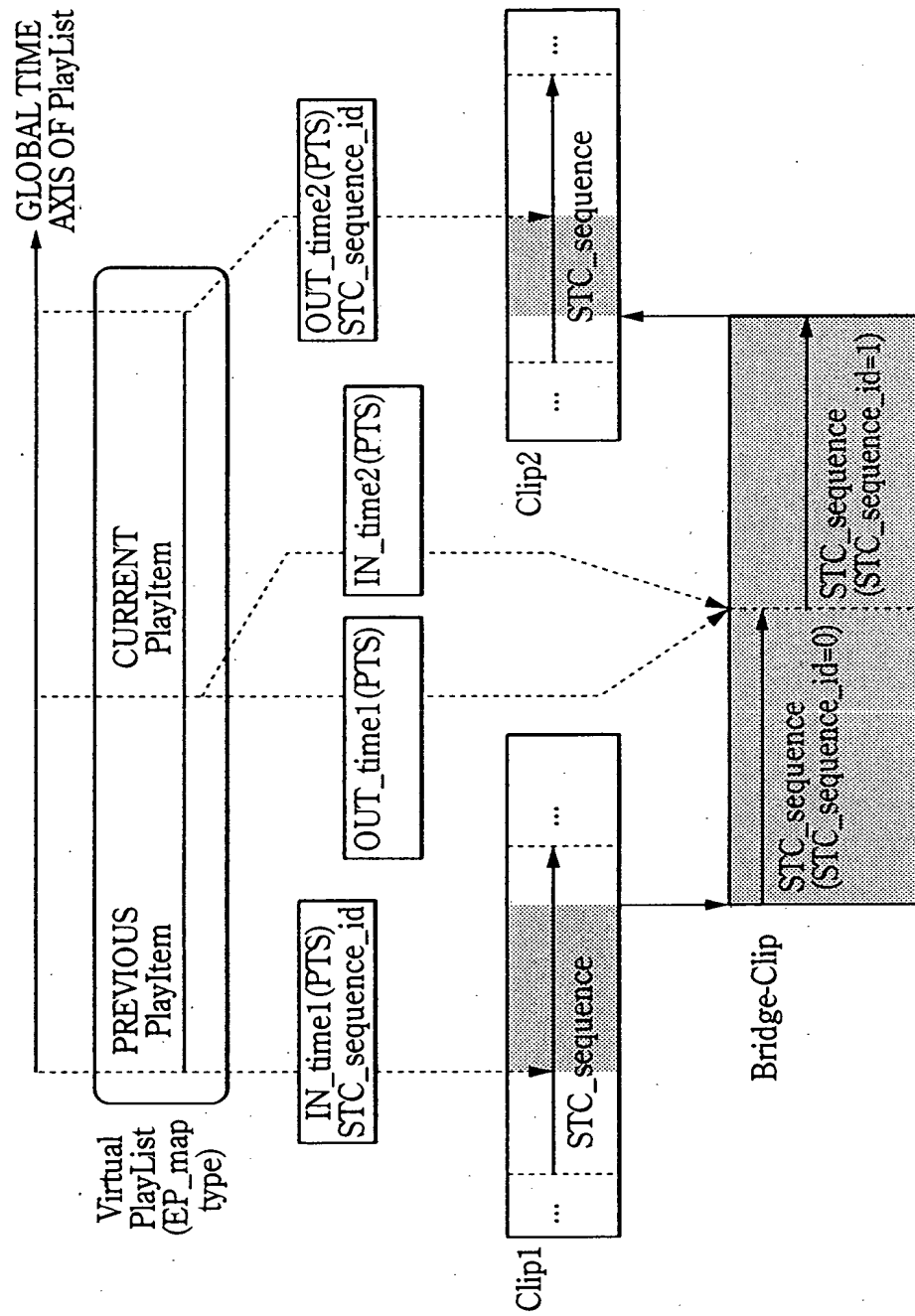
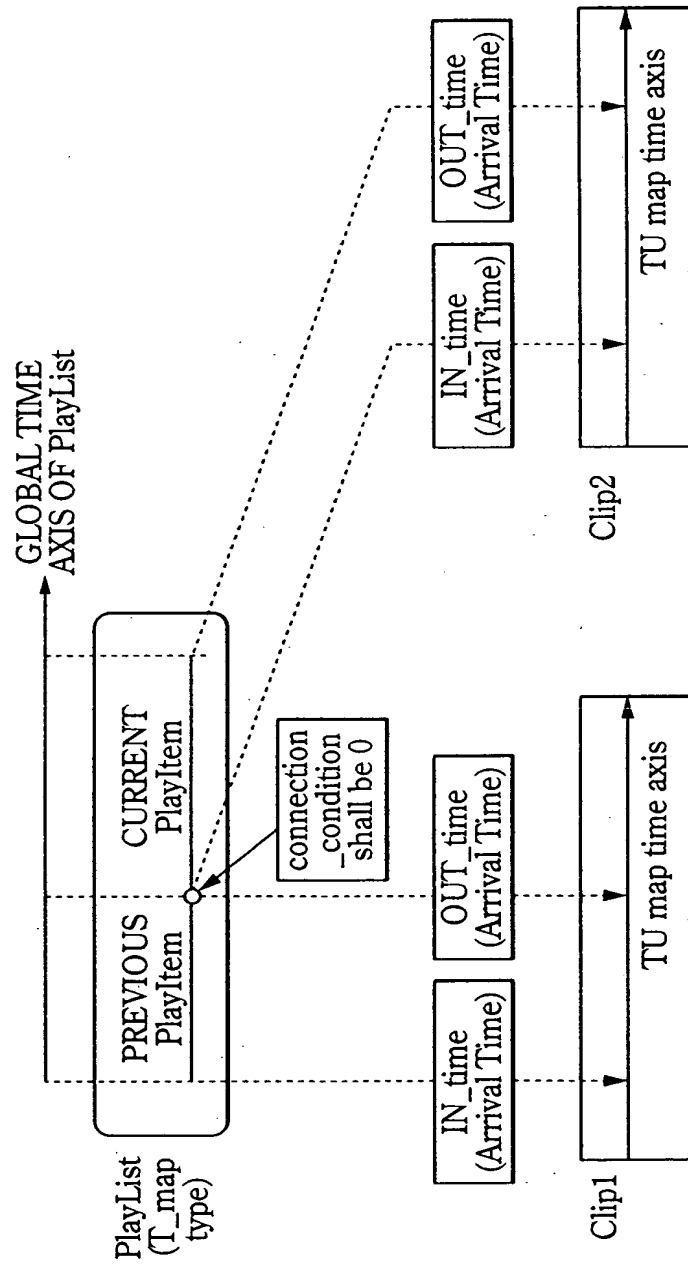


FIG.30

**FIG.31**

SYNTAX	NUMBER OF BYTES	ABBREVIATION
PlayItem(){		
<b>Clip_information_file_name</b>	8*10	bslbf
reserved	24	bslbf
<b>STC_sequence_id</b>	8	uimsbf
<b>IN_time</b>	32	uimsbf
<b>OUT_time</b>	32	uimsbf
reserved	14	bslbf
<b>connection_condition</b>	2	bslbf
if (<Virtual PlayList>){		
if ( <i>connection_condition</i> =='10'){		
<b>BridgeSequenceInfo()</b>		
}		
}		
}		

FIG.32

CPI_type in the PlayList()	SEMANTICS OF IN_time
EP_map type	IN_time MUST INDICATE UPPER 32 BITS OF 33 BIT LENGTH CORRESPONDING TO FIRST PRESENTATION UNIT IN PlayItem
TU_map type	IN_time MUST BE TIME ON TU_map_time_axis, AND MUST BE ROUNDED TO time_unit PRECISION. IN-time IS CALCULATED BY FOLLOWING EQUATION:  $\text{IN\_time} = \text{TU\_start\_time} \% 2^{32}$

**FIG.33**

CPI_type in the PlayList()	SEMANTICS OF OUT_time
EP_map type	<p>OUT_time MUST INDICATE UPPER 32 BITS OF THE VALUE OF Presentation_end_TS CALCULATED BY FOLLOWING EQUATION:</p> $\text{Presentation\_end\_TS} = \text{PTS\_out} + \text{AU\_duration}$ <p>WHERE PTS_out IS 33-BIT LONG PTS CORRESPONDING TO LAST PRESENTATION UNIT IN PlayItem. AU_duration IS 90 kHz-DISPLAY TIME OF LAST PRESENTATION UNIT.</p>
TU_map type	<p>OUT_time MUST BE TIME ON TU_map_time_axis AND BE ROUNDED TO time_unit PRECISION. OUT_time IS CALCULATED BY FOLLOWING EQUATION:</p> $\text{OUT\_time} = \text{TU\_start\_time} \% 2^{32}$

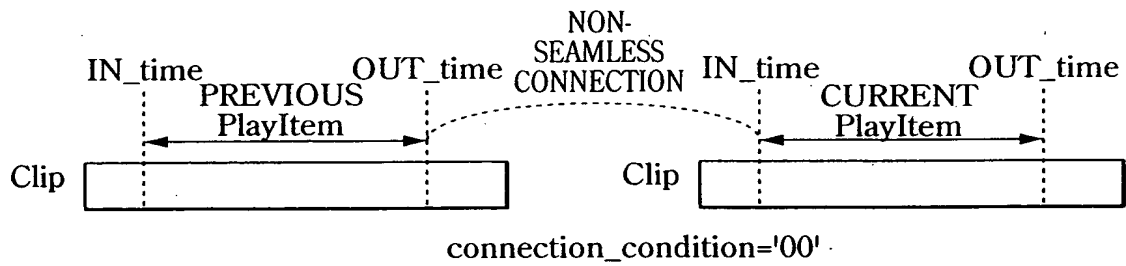
FIG.34



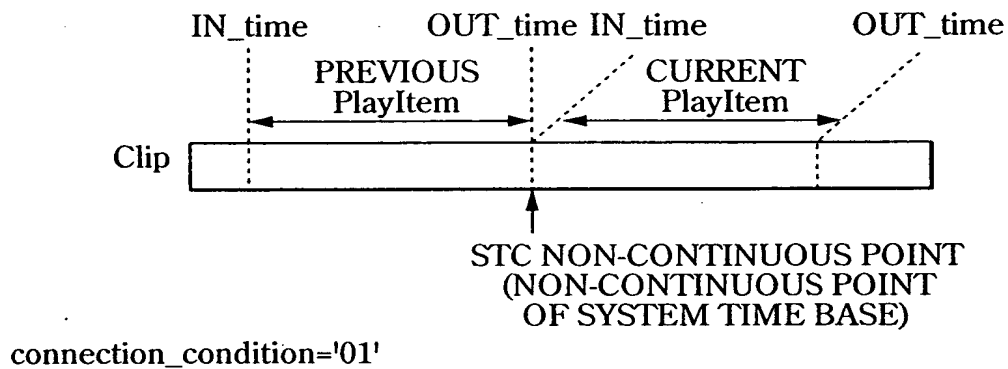
connection _condition	MEANING
00	<ul style="list-style-type: none"> <li>• CONNECTION OF PREVIOUS PlayItem TO CURRENT PlayItem IS NOT SURE AS TO SEAMLESS REPLAY.</li> <li>• IF CPI_type OF PlayList IS TU_map type, THIS VALUE MUST BE SET IN connection_condition.</li> </ul>
01	<ul style="list-style-type: none"> <li>• THIS STATE IS ALLOWED ONLY WHEN CPI_type OF PlayList IS EP_map type.</li> <li>• PREVIOUS PlayItem AND CURRENT PlayItem INDICATE DIVISION BECAUSE OF NON-CONTINUOUS POINT OF SYSTEM TIMEBASE (STC BASE).</li> </ul>
10	<ul style="list-style-type: none"> <li>• THIS STATE IS ALLOWED ONLY WHEN CPI_type OF PlayList IS EP_map type.</li> <li>• THIS STATE IS ALLOWED ONLY FOR Virtual PlayList.</li> <li>• CONNECTION OF PREVIOUS PlayItem TO CURRENT PlayItem IS SURE AS TO SEAMLESS REPLAY.</li> <li>• PREVIOUS PlayItem IS CONNECTED TO CURRENT PlayItem USING BridgeSequence. DVR MPEG-2 TRANSPORT STREAM MUST OBEY DVR-STD AS LATER DESCRIBED.</li> </ul>
11	<ul style="list-style-type: none"> <li>• THIS STATE IS ALLOWED ONLY WHEN CPI_type OF PlayList IS EP_map type.</li> <li>• CONNECTION OF PREVIOUS PlayItem TO CURRENT Play Item IS SURE AS TO SEAMLESS REPLAY.</li> <li>• PREVIOUS PlayItem IS CONNECTED TO CURRENT PlayItem WITHOUT USING BridgeSequence. DVR MPEG-2 TRANSPORT STREAM MUST OBEY DVR-STD AS LATER DESCRIBED.</li> </ul>

FIG.35

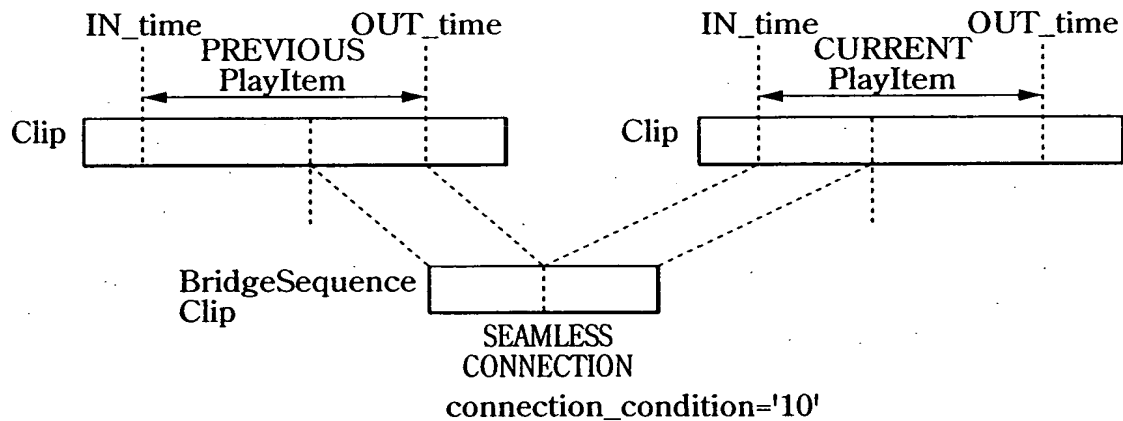
34/118



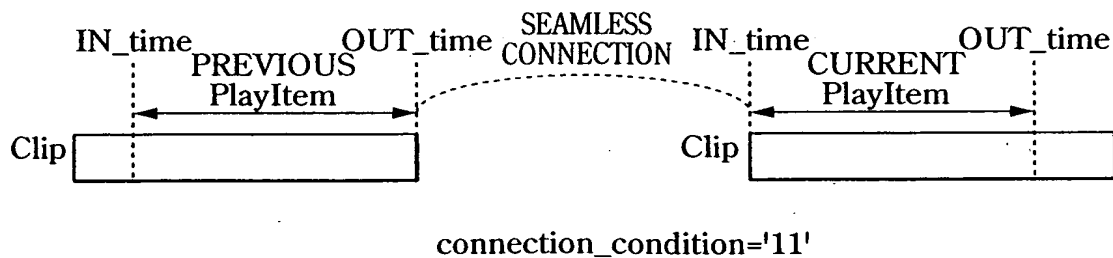
**FIG.36A**



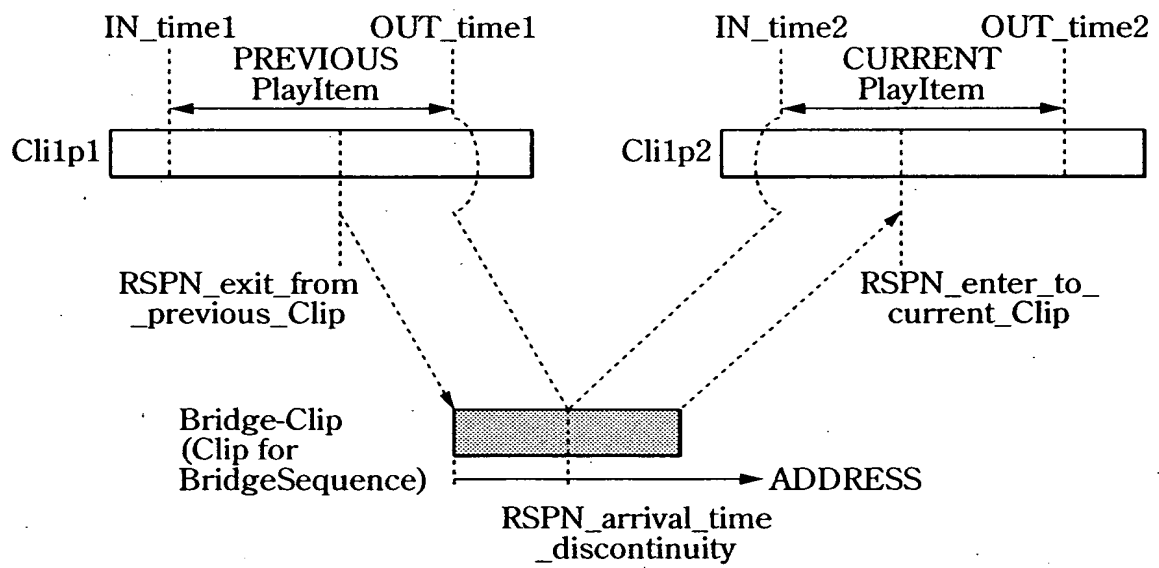
**FIG.36B**



**FIG.36C**

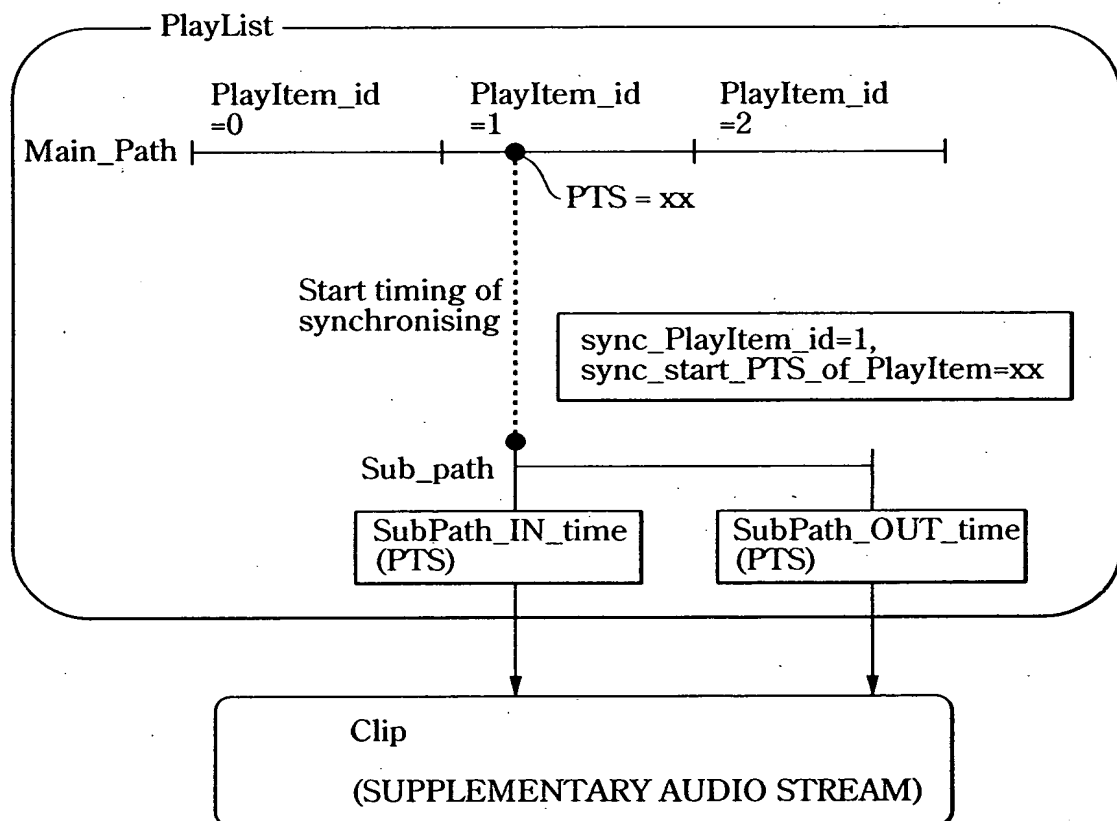


**FIG.36D**

**FIG.37**

SYNTAX	NUMBER OF BYTES	ABBREVIATION
BridgeSequenceInfo() {		
Bridge_Clip_information_file_name	8*10	bslbf
RSPN_exit_from_previous_Clip	32	uimsbf
RSPN_enter_to_current_Clip	32	uimsbf
}		

FIG.38

**FIG.39**

SYNTAX	NUMBER OF BYTES	ABBREVIATION
SubPlayItem(){		
<b>Clip_Information_file_name</b>	8*10	bslbf
<b>SubPath_type</b>	8	bslbf
<b>sync_PlayItem_id</b>	8	uimsbf
<b>sync_start_PTS_of_PlayItem</b>	32	uimsbf
<b>SubPath_IN_time</b>	32	uimsbf
<b>SubPath_OUT_time</b>	32	uimsbf
}		

**FIG.40**

SubPath_type	MEANING
0x00	Auxiliary audio steam path
0x01-0xff	reserved

**FIG.41**

SYNTAX	NUMBER OF BYTES	ABBREVIATION
PlayListMark(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_PlayList_marks</b>	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
<b>reserved</b>	8	bslbf
<b>mark_type</b>	8	bslbf
<b>mark_time_stamp</b>	32	uimsbf
<b>PlayItem_id</b>	8	uimsbf
<b>reserved</b>	24	uimsbf
<b>character_set</b>	8	bslbf
<b>name_length</b>	8	uimsbf
<b>mark_name</b>	8*256	bslbf
<b>ref_thumbnail_index</b>	16	uimsbf
}		
}		

FIG.42



Mark_type	MEANING	COMMENT
0x00	resume-mark	REPLAY RESUME POINT. THE NUMBER OF REPLAY RESURE POINTS DEFINED IN PlayListMark() MUST BE 0 OR 1.
0x01	book-mark	REPLAY ENTRY POINT OF PlayList. THIS MARK CAN BE SET BY USER AND USED AS MARK SPECIFYING START POINT OF FAVORITE SCENE.
0x02	skip-mark	SKIP MARK POINT. PLAYER SKIPS PROGRAM FROM THIS POINT TO THE END OF PROGRAM. THE NUMBER OF SKIP MARK POINTS DEFINED IN PlayListMark() MUST BE 0 RO 1.
0x03-0x8F	reserved	
0x90-0xFF	reserved	Reserved for ClipMark()

FIG.43

CPI_type in the PlayList()	SEMANTICS OF mark_time_stamp
EP_map type	mark_time_stamp MUST INDICATE UPPER 32 BITS OF 33 BIT LENGTH PTS CORRESPONDING TO PRESENTATION UNIT REFERENCED BY MARK.
TU_map type	mark_time_stamp MUST BE TIME ON TU_map_time_axis AND MUST BE ROUNDED TO time_unit PRECISION. mark_time_stamp IS CALCULATED BY FOLLOWING EQUATION:  $\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$

FIG.44

SYNTAX	NUMBER OF BYTES	ABBREVIATION
<b>zzzzz.cpi {</b>		
<b>STC_Info_Start_address</b>	32	uimsbf
<b>ProgramInfo_Start_address</b>	32	uimsbf
<b>CPI_Start_address</b>	32	uimsbf
<b>ClipMark_Start_address</b>	32	uimsbf
<b>MakersPrivateData_Start_address</b>	32	uimsbf
reserved	96	bslbf
<b>ClipInfo()</b>		
for (i=0;i<N1;i++){		
<b>padding_word</b>	16	bslbf
}		
<b>STC_Info()</b>		
for (i=0;i<N2;i++){		
<b>padding_word</b>	16	bslbf
}		
<b>ProgramInfo()</b>		
for (i=0;i<N3;i++){		
<b>padding_word</b>	16	bslbf
}		
<b>CPI()</b>		
for (i=0;i<N4;i++){		
<b>padding_word</b>	16	bslbf
}		
<b>ClipMark()</b>		
for (i=0;i<N5;i++){		
<b>padding_word</b>	16	bslbf
}		
<b>MakersPrivateData()</b>		
<b>}</b>		

FIG.45

SYNTAX	NUMBER OF BYTES	ABBREVIATION
ClipInfo(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>Clip_stream_type</b>	8	bslbf
<b>offset_SPN</b>	32	uimsbf
<b>TS_recording_rate</b>	24	uimsbf
reserved	8	bslbf
<b>record_time_and_date</b>	4*14	bslbf
reserved	8	bslbf
<b>duration</b>	4*6	bslbf
reserved	7	bslbf
<b>time_controlled_flag</b>	1	bslbf
<b>TS_average_rate</b>	24	uimsbf
<i>if (Clip_stream_type==1) // Bridge-Clip AV stream</i>		
<b>RSPN_arrival_time_discontinuity</b>	32	uimsbf
else		
reserved	32	bslbf
<b>reserved_for_system_use</b>	144	bslbf
reserved	11	bslbf
<b>is_format_identifier_valid</b>	1	bslbf
<b>is_original_network_ID_valid</b>	1	bslbf
<b>is_transport_stream_ID_valid</b>	1	bslbf
<b>is_service_ID_valid</b>	1	bslbf
<b>is_country_code_valid</b>	1	bslbf
<b>format_identifier</b>	32	bslbf
<b>original_network_ID</b>	16	uimsbf
<b>transport_stream_ID</b>	16	uimsbf
<b>service_ID</b>	16	uimsbf
<b>country_code</b>	24	bslbf
<b>stream_format_name</b>	16*8	bslbf
<b>reserved_for_fortune_use</b>	256	bslbf
}		

FIG.46

45/118

Clip_stream_type	MEANING
0	Clip AV STREAM
1	Bridge-Clip AV STREAM
2-255	Reserved

**FIG.47**

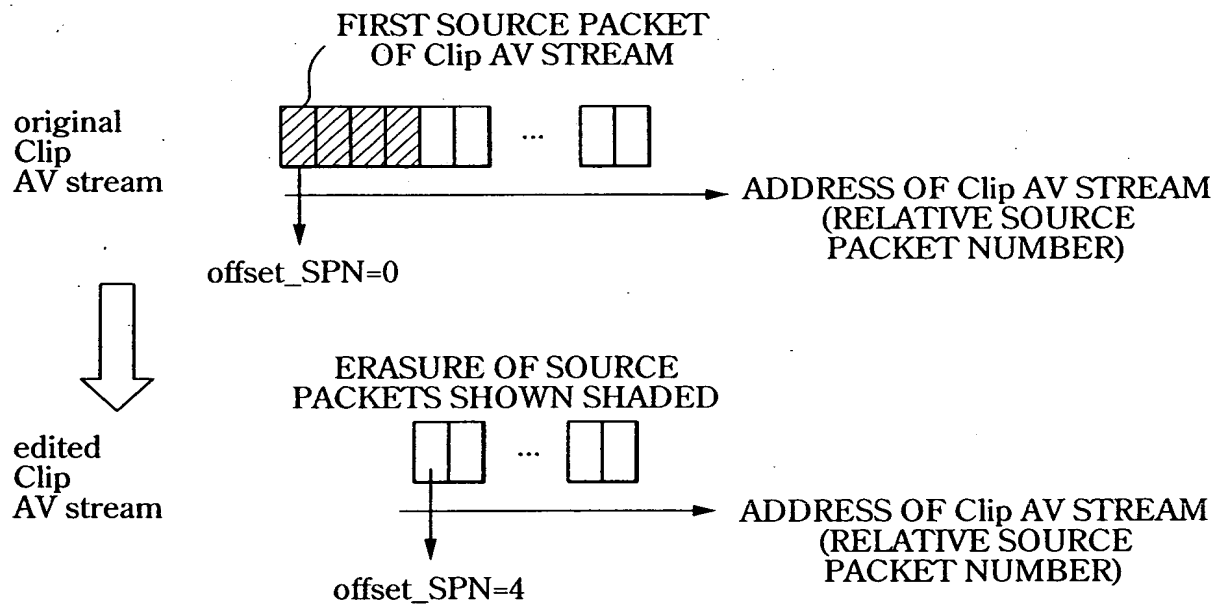


FIG.48

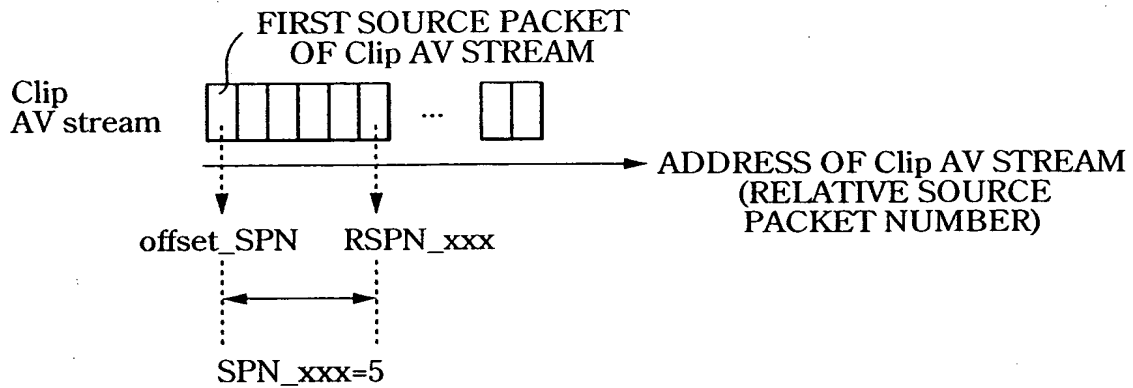


FIG.49

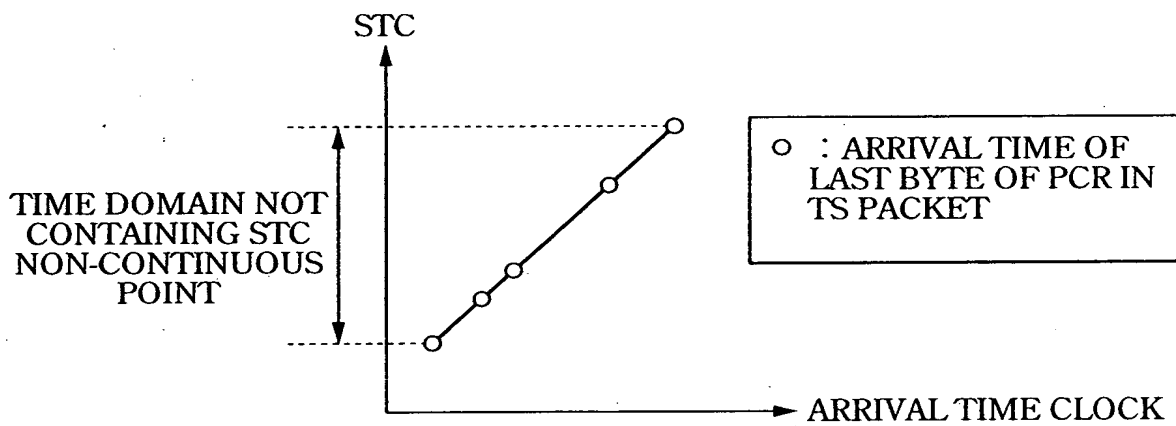


FIG.50A

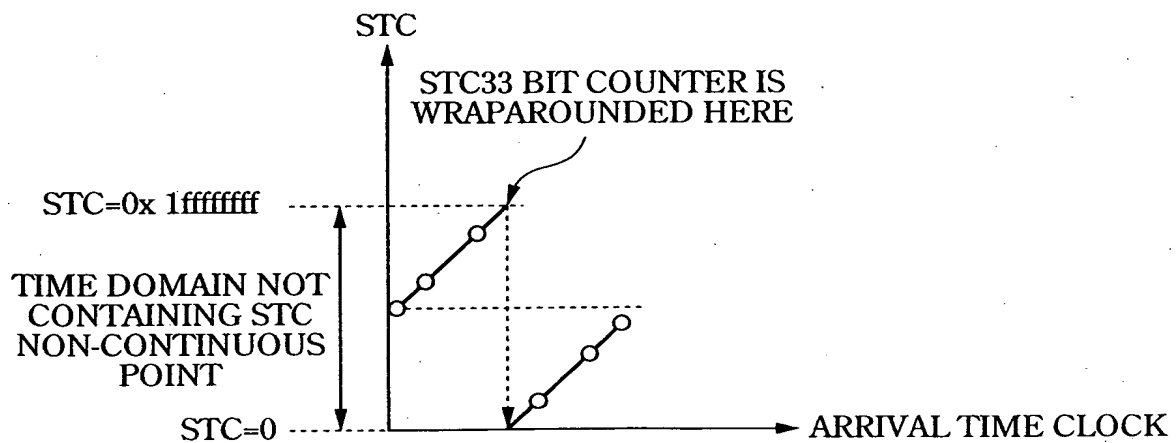


FIG.50B

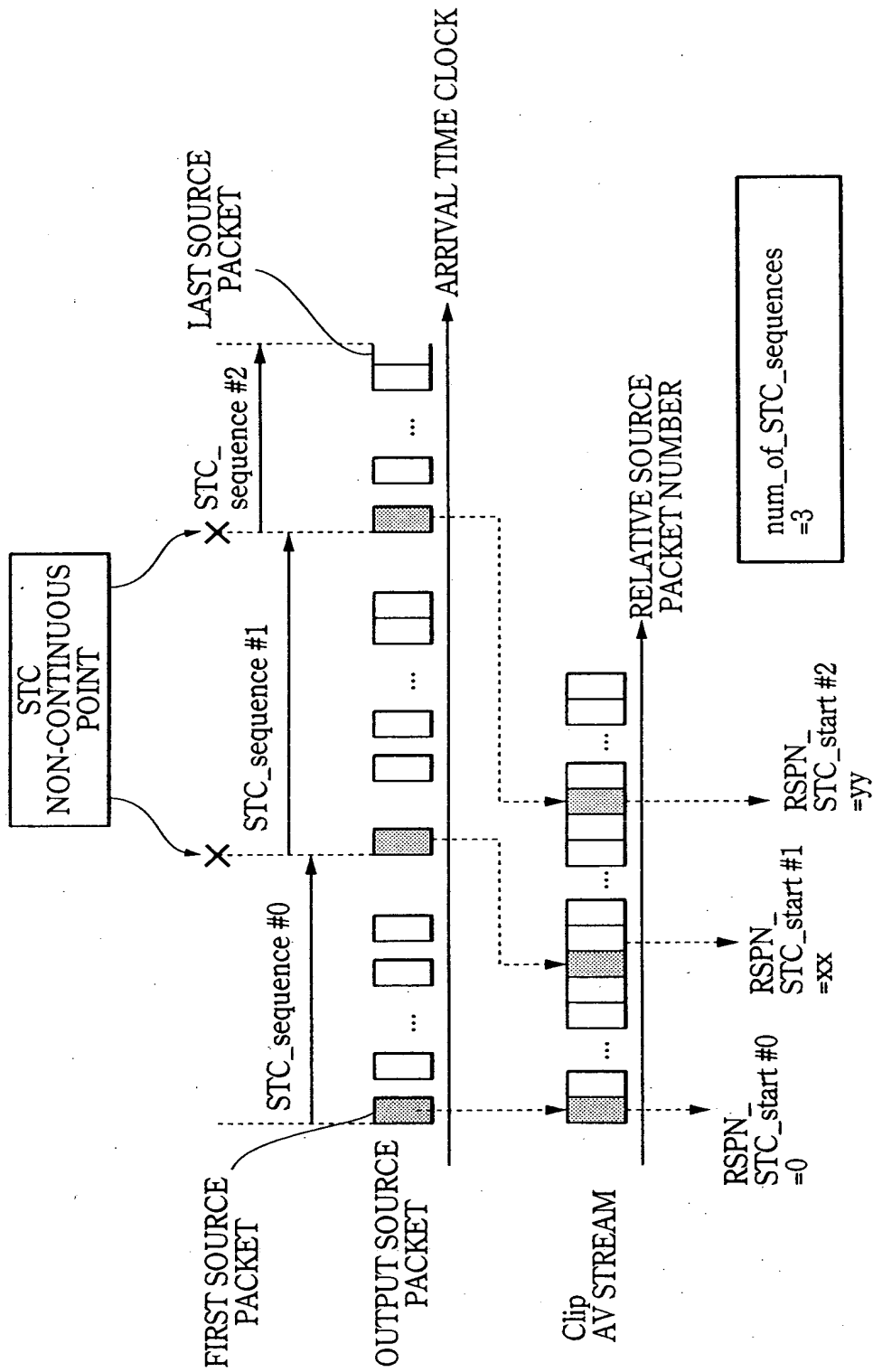


FIG.51



SYNTAX	NUMBER OF BYTES	ABBREVIATION
<b>STC_Info()</b> {		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
if (length !=0){		
reserved	8	bslbf
<b>num_of_STC_sequences</b>	8	uimsbf
for ( <i>STC_sequence_id</i> =0; <i>STC_sequence_id</i> < <i>num_of_STC_sequences</i> ; <i>STC_sequence_id</i> ++){		
resereved	32	bslbf
<b>RSPN_STC_start</b>	32	uimsbf
}		
}		
}		

FIG.52

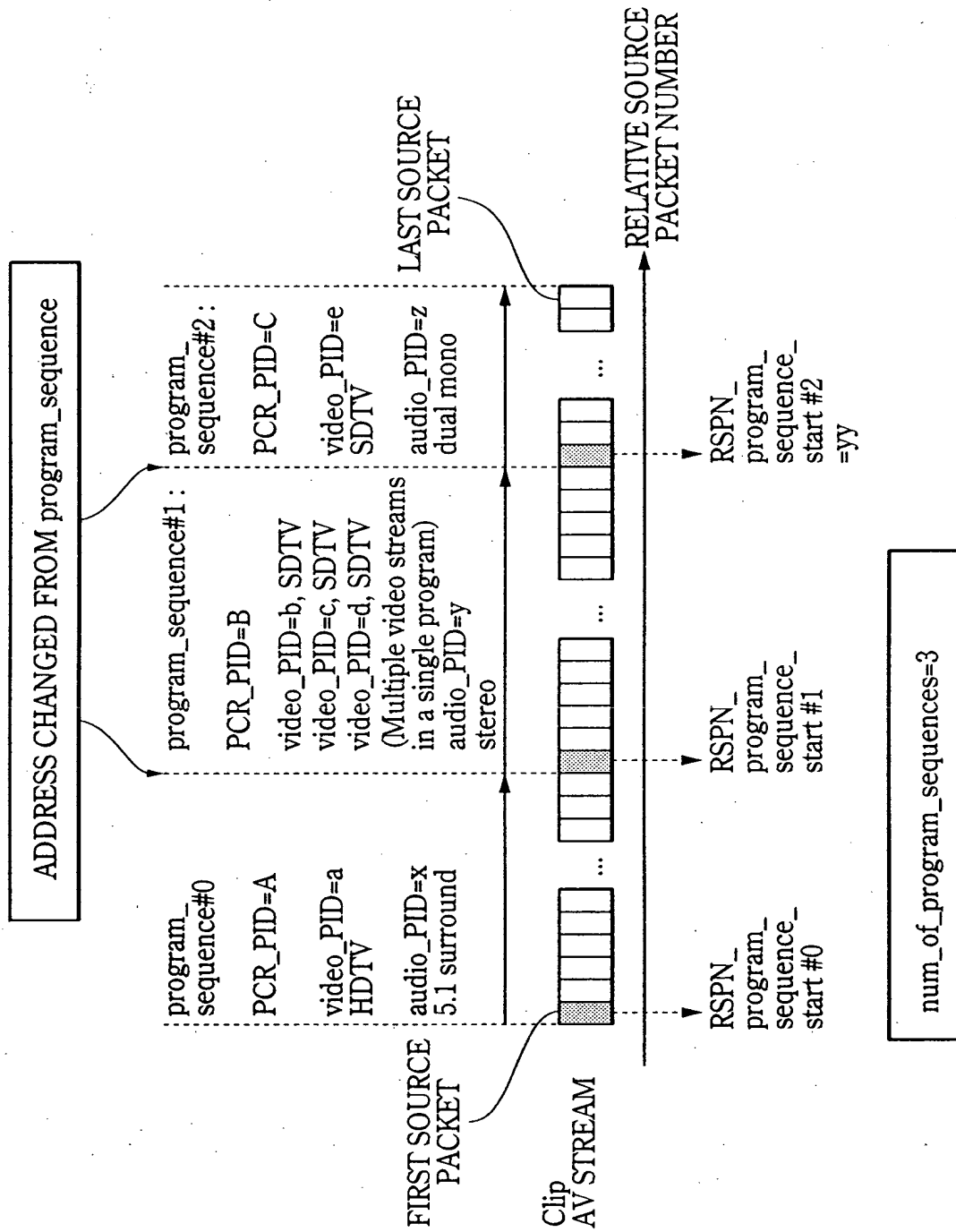


FIG.53

SYNTAX	NUMBER OF BYTES	ABBREVIATION
<b>ProgramInfo()</b>		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
if (length !=0){		
reserved	8	bslbf
<b>number_of_program_sequences</b>	8	uimsbf
for (i=0;i< <b>number_of_program_sequences</b> ;i++){		
<b>RSPN_program_sequence_start</b>	32	uimsbf
reserved	48	bslbf
<b>PCR_PID</b>	16	bslbf
<b>number_of_videos</b>	8	uimsbf
<b>number_of_audios</b>	8	uimsbf
for (k=0;k< <b>number_of_videos</b> ;k++){		
<b>video_stream_PID</b>	16	bslbf
<b>VideoCodingInfo()</b>		
}		
for (k=0;k< <b>number_of_audios</b> ;k++){		
<b>audio_stream_PID</b>	16	bslbf
<b>AudioCodingInfo()</b>		
}		
}		
}		
}		

FIG.54

SYNTAX	NUMBER OF BYTES	ABBREVIATION
VideoCodingInfo() {		
<b>video_format</b>	8	uimsbf
<b>frame_rate</b>	8	uimsbf
<b>display_aspect_ratio</b>	8	uimsbf
<b>reserved</b>	8	bslbf
}		

FIG.55

video_format	MEANING
0	480i
1	576i
2	480p(including 640×480p format)
3	1080i
4	720p
5	1080p
6-254	reserved
255	No information

**FIG.56**

frame_rate	MEANING
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97..)
5	30
6	50
7	60 000/1001 (59.94..)
8	60
9-254	reserved
255	No information

**FIG.57**

display_aspect_ratio	MEANING
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ration
4-254	reserved
255	No information

**FIG.58**

SYNTAX	NUMBER OF BYTES	ABBREVIATION
AudioCodingInfo() {		
<b>audio_format</b>	8	uimsbf
<b>audio_component_type</b>	8	uimsbf
<b>sampling_frequency</b>	8	uimsbf
reserved	8	bslbf
}		

FIG.59

audio_coding	MEANING
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

**FIG.60**



audio_component_type	MEANING
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

**FIG.61**

sampling_frequency	MEANING
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

**FIG.62**

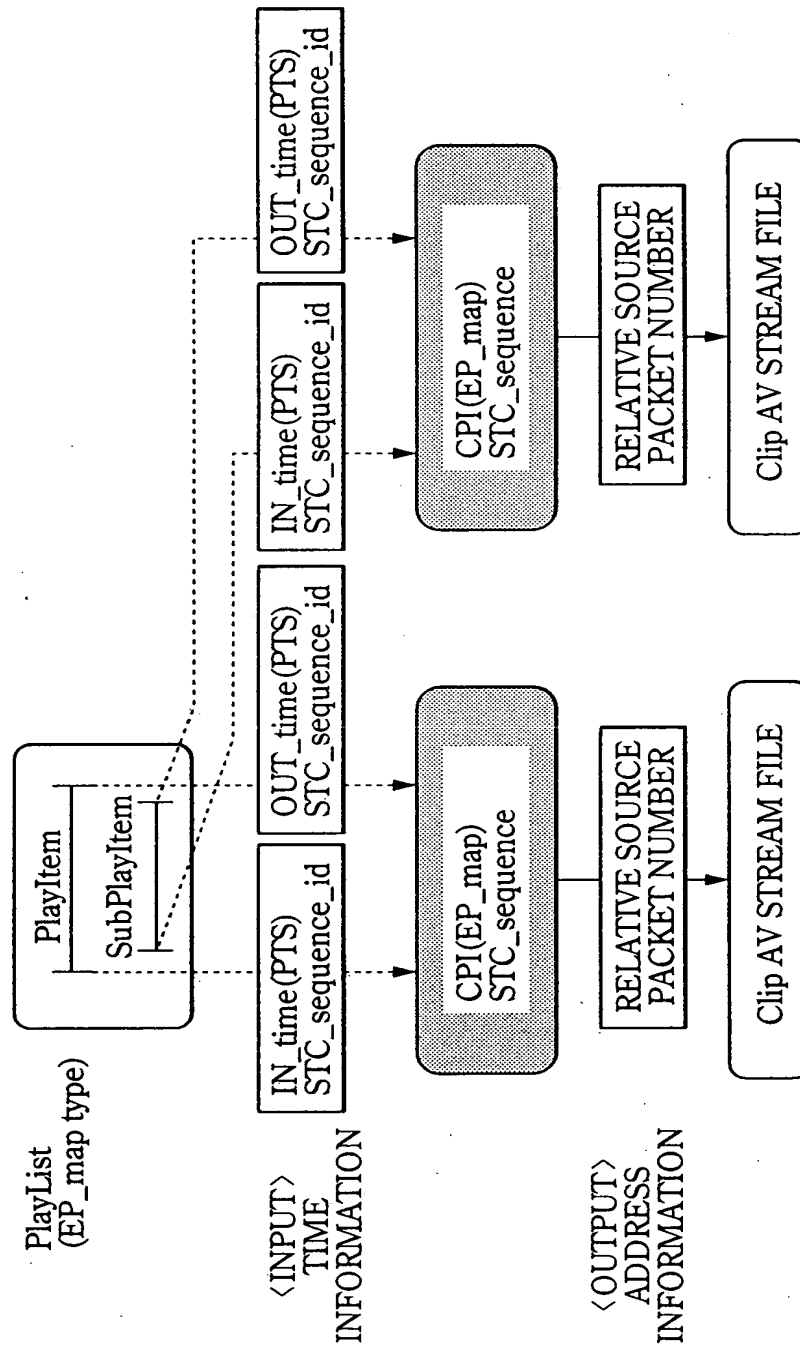
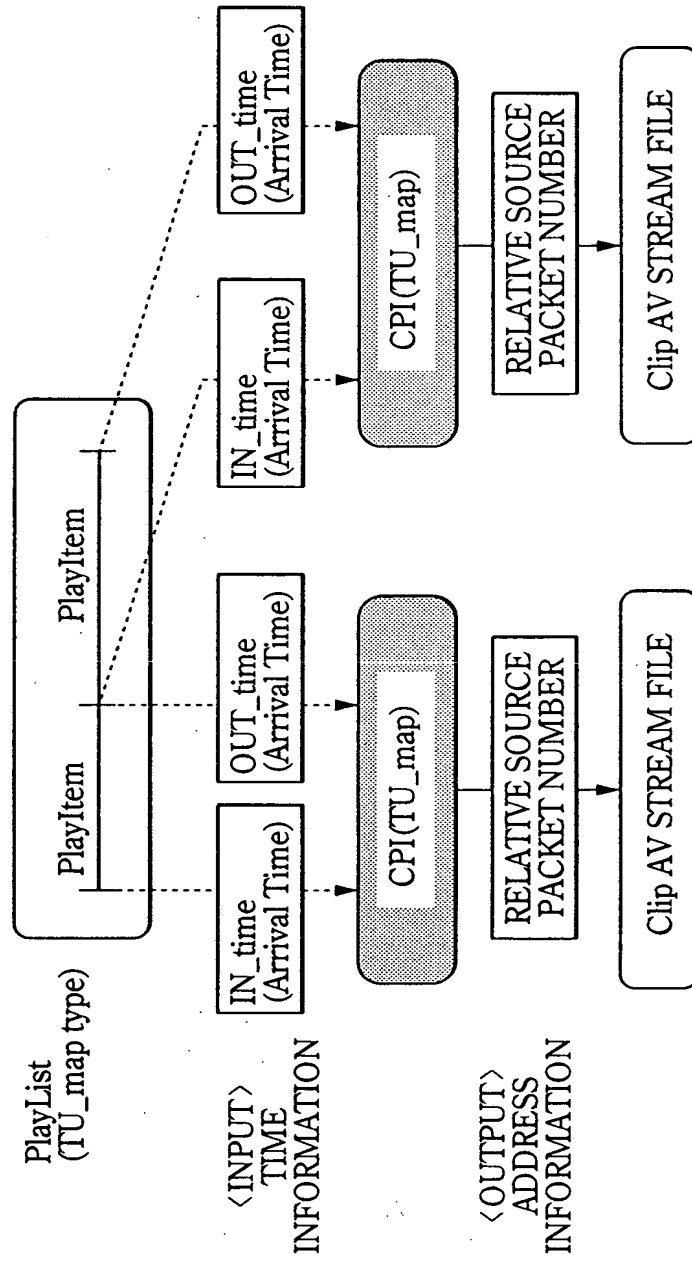


FIG. 63

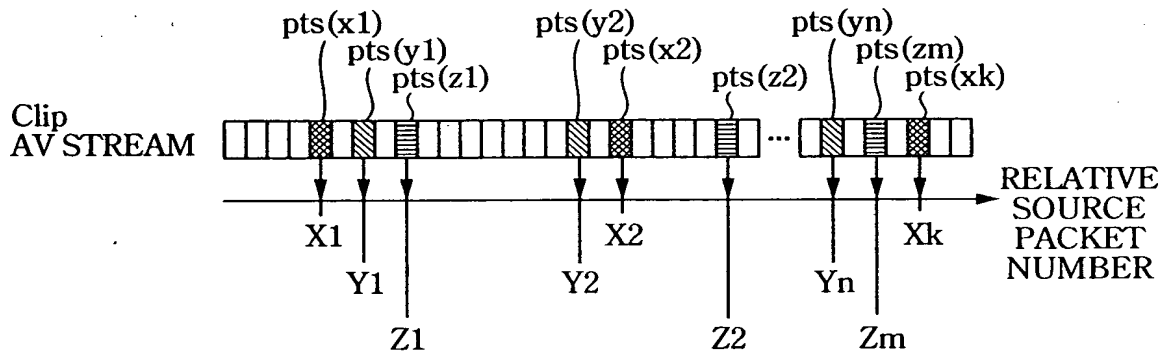
**FIG.64**

SYNTAX	NUMBER OF BYTES	ABBREVIATION
CPI(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
reserved	15	bslbf
<b>CPI_type</b>	1	bslbf
if (CPI_type==0)		
<b>EP_map()</b>		
else		
<b>TU_map()</b>		
}		

FIG.65

CPI_type	MEANING
0	EP map type
1	TU map type

FIG.66



- : SOURCE PACKET CONTAINING FIRST BYTE OF SEQUENCE  
 HEADER video\_PID=x  
 ▨ : SOURCE PACKET CONTAINING FIRST BYTE OF SEQUENCE  
 HEADER video\_PID=y  
 ▩ : SOURCE PACKET CONTAINING FIRST BYTE OF SEQUENCE  
 HEADER video\_PID=z

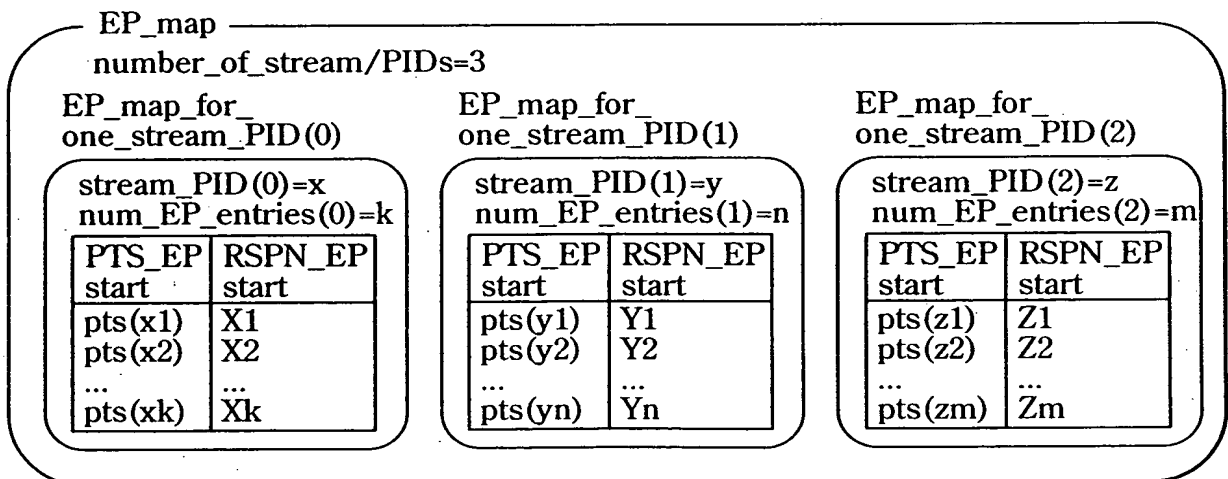
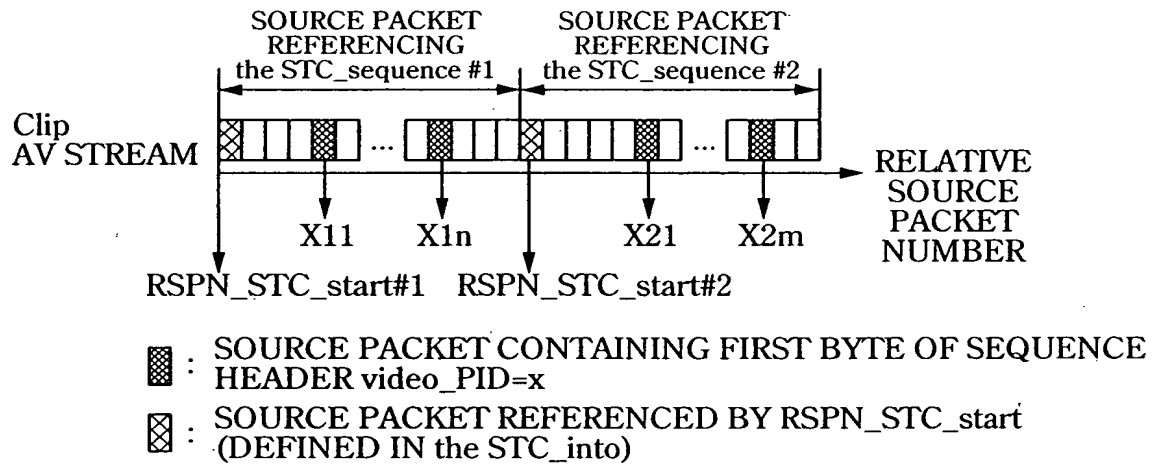


FIG.67



EP\_map\_for\_one\_stream\_PID  
video\_PID=x

PTS_EP start	RSPN_EP start	
pts(x11)	X11	DATA BELONGING TO STC_sequence #1
...	...	
pts(x1n)	X1n	
		→ boundary
pts(x21)	X21	DATA BELONGING TO STC_sequence #2
...	...	
pts(x2m)	X2m	

RSPN\_STC\_start #2 < X21

FIG.68

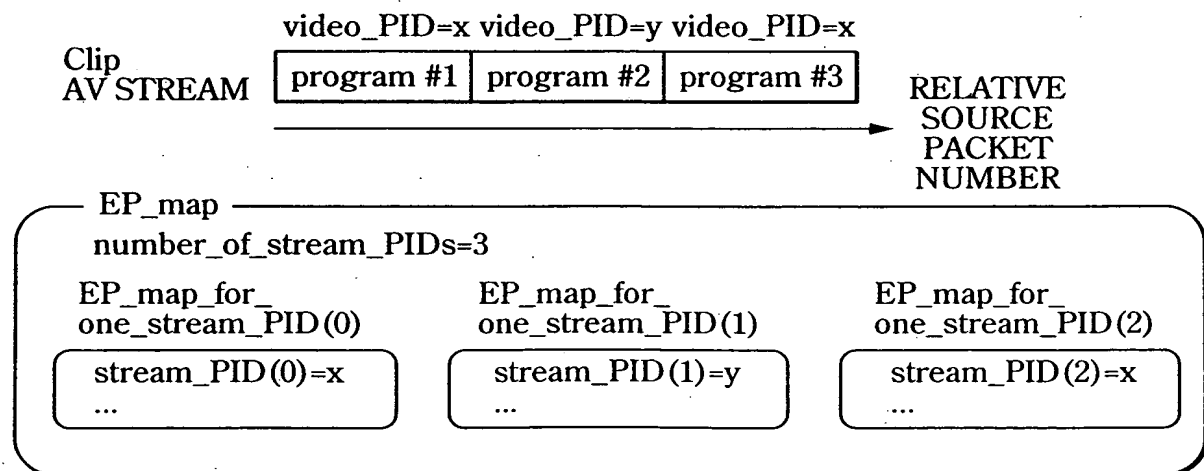


FIG.69

SYNTAX	NUMBER OF BYTES	ABBREVIATION
EP_map(){		
reserved	12	bslbf
EP_type	4	uimsbf
number_of_stream_PIDs	16	uimsbf
for (k=0;k<number_of_stream_PIDs;k++){		
stream_PID(k)	16	bslbf
num_EP_entries(k)	32	uimsbf
EP_map_for_one_stream_PID_Start_address(k)	32	uimsbf
}		
for (i=0;i<X;i++){		
padding_word	16	bslbf
}		
for (k=0;k<number_of_stream_PIDs;k++){		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (i=0;i<Y;i++){		
padding_word	16	bslbf
}		
}		
}		

FIG.70

**64/118**

EP_type	MEANING
0	video
1	audio
2-15	reserved

**FIG.71**



SYNTAX	NUMBER OF BYTES	ABBREVIATION
EP_map_for_one_stream_PID( <i>N</i> ) {		
for (i=0;i< <i>N</i> ;i++){		
PTS_EP_start	32	uimsbf
RSPN_EP_start	32	uimsbf
}		
}		

FIG.72

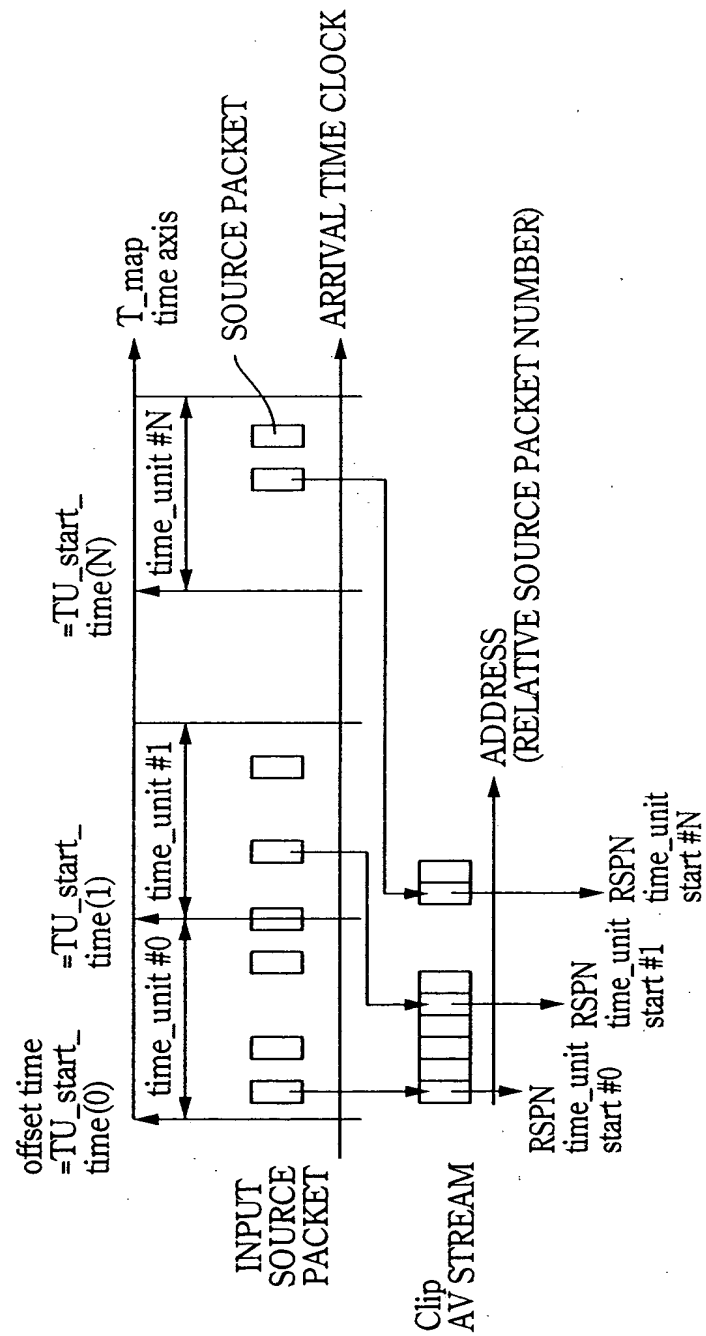


FIG.73

SYNTAX	NUMBER OF BYTES	ABBREVIATION
TU_map() {		
offset_time	32	bslbf
time_unit_size	32	uimsbf
number_of_time_unit_entries	32	uimsbf
for (k=0;k<number_of_time_unit_entries;k++)		
RSPN_time_unit_start	32	uimsbf
}		

FIG.74

SYNTAX	NUMBER OF BYTES	ABBREVIATION
ClipMark(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_Clip_marks</b>	16	uimsbf
for (i=0; i<number_of_clip_marks; i++){		
reserved	8	bslbf
<b>mark_type</b>	8	bslbf
<b>mark_time_stamp</b>	32	uimsbf
<b>STC_sequence_id</b>	8	uimsbf
reserved	24	bslbf
<b>character_set</b>	8	bslbf
<b>name_length</b>	8	uimsbf
<b>mark_name</b>	8*256	bslbf
<b>ref_thumbnail_index</b>	16	uimsbf
}		
}		

FIG.75

Mark_type	MEANING	COMMENT
0x00-0x8F	reserved	Reserved for PlayListMark0
0x90	Event-start mark	MARK POINT INDICATING PROGRAM START POINT
0x91	Local event-start mark	MARK POINT INDICATING LOCAL SCENE IN PROGRAM
0x92	Scene-start mark	MARK POINT SHOWING SCENE CHANGE POINT
0x93-0xFF	reserved	

**FIG.76**

CPI_type in the PlayList()	SEMANTICS OF mark_time_stamp
EP_map type	mark_time_stamp MUST INDICATE UPPER 32 BITS OF 33 BIT LENGTH PTS CORRESPONDING TO PRESENTATION UNIT REFERENCED BY MARK.
TU_map type	mark_time_stamp MUST BE TIME ON TU_map_time_axis AND MUST BE ROUNDED TO time_unit PRECISION. mark_time_stamp IS CALCULATED BY FOLLOWING EQUATION:  $\text{mark\_time\_stamp} = \text{TU\_start\_time} \% 2^{32}$

FIG.77

SYNTAX	NUMBER OF BYTES	ABBREVIATION
ClipMark(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_Clip_marks</b>	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bslbf
<b>mark_type</b>	8	bslbf
<b>reserved_for_MakerID</b>	16	bslbf
<i>mark_entry()</i>		
<i>representative_picture_entry()</i>		
<b>ref_thumbnail_index</b>	16	uimsbf
}		
}		

FIG.78

Mark_type	MEANING	COMMENT
0x00-0x8F	reserved	Reserved for PlayListMark()
0x90	Event-start mark	MARK POINT INDICATING PROGRAM START POINT
0x91	Local event-start mark	MARK POINT INDICATING LOCAL SCENE IN PROGRAM
0x92	Scene-start mark	MARK POINT INDICATING SCENE START POINT
0x93	Scene-end mark	MARK POINT INDICATING SCENE END POINT
0x94	CM-start mark	MARK POINT INDICATING CM START POINT
0x95	CM-end mark	MARK POINT INDICATING CM END POINT
0x96-0xBF	DVR FORMAT IS RESERVED FOR FUTURE EXTENSION OF ClipMark	
0xC0-0xFF	ALLOCATBLE TO MARK USED IN MAKER-UNIQUE APPLICCATION	

FIG.79

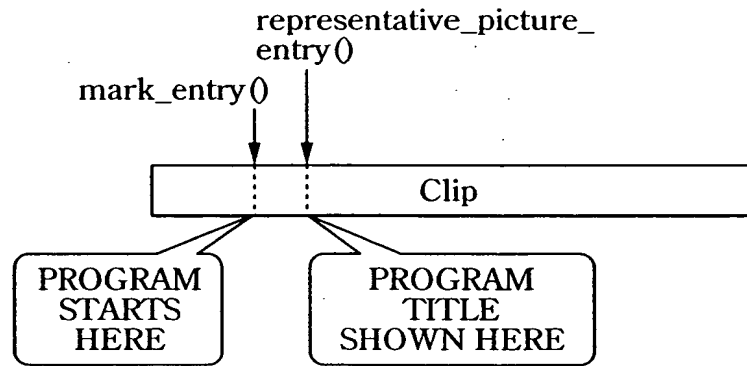


FIG.80

SYNTAX	NUMBER OF BYTES	ABBREVIATION
<b>mark_entry()/representative_picture_entry(){</b>		
<b>mark_time_stamp</b>	32	uimsbf
<b>STC_sequence_id</b>	8	uimsbf
<b>reserved</b>	24	bslbf
<b>}</b>		

FIG.81

SYNTAX	NUMBER OF BYTES	ABBREVIATION
<b>mark_entry()/representative_picture_entry(){</b>		
<b>RSPN_ref_EP_start</b>	32	uimsbf
<b>offset_num_pictures</b>	32	uimsbf
<b>}</b>		

FIG.82



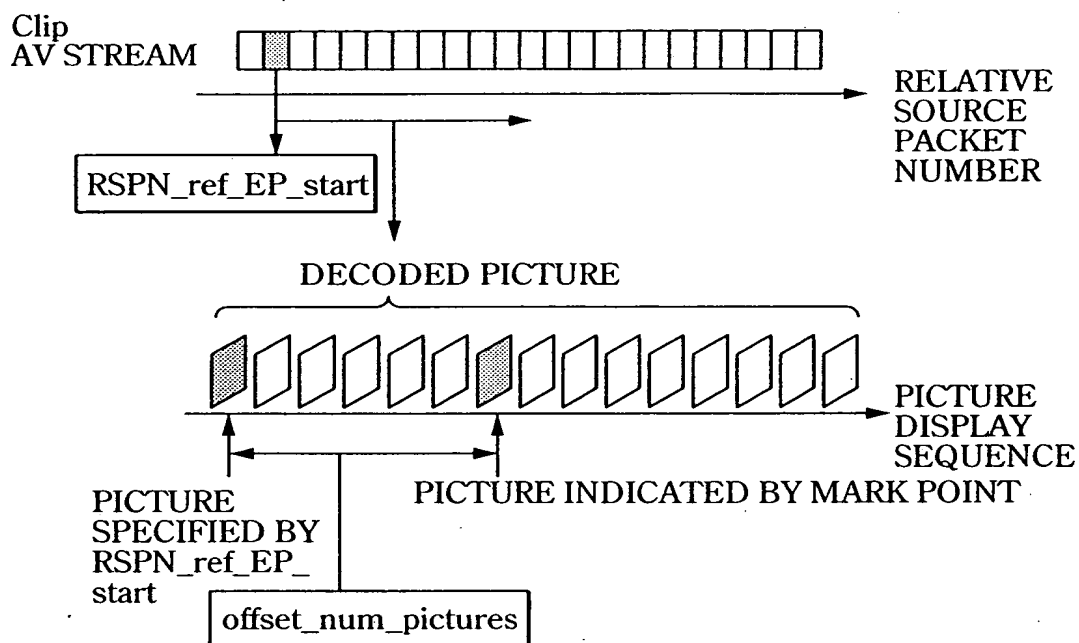


FIG.83

SYNTAX	NUMBER OF BYTES	ABBREVIATION
mark_entry()/representative_picture_entry(){		
<b>RSPN_mark_point</b>	32	uimsbf
}		

FIG.84

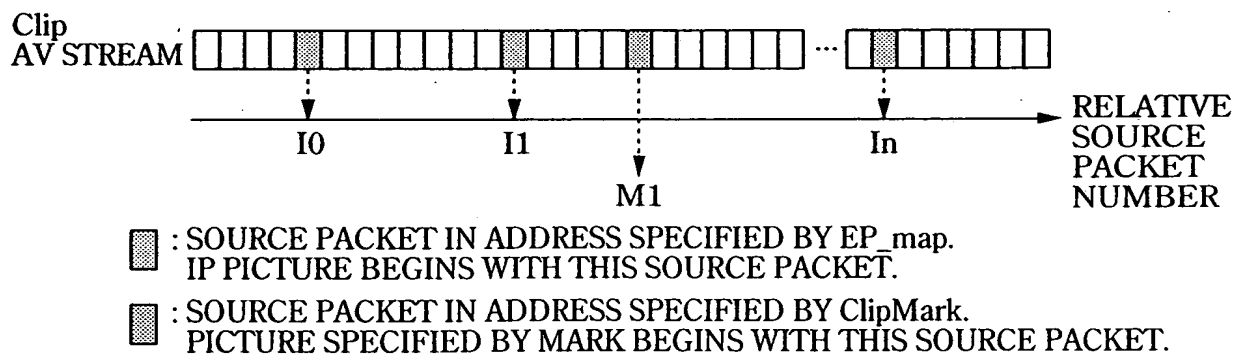


FIG.85

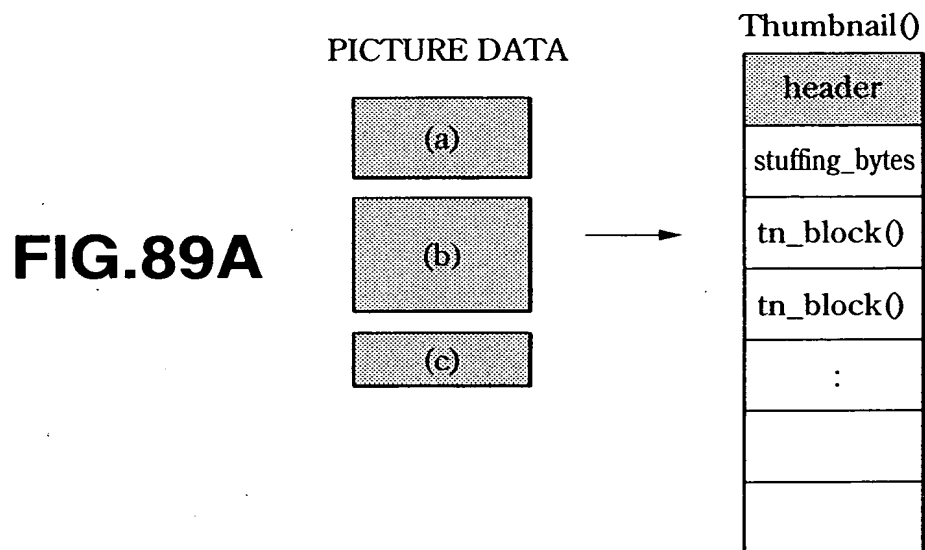
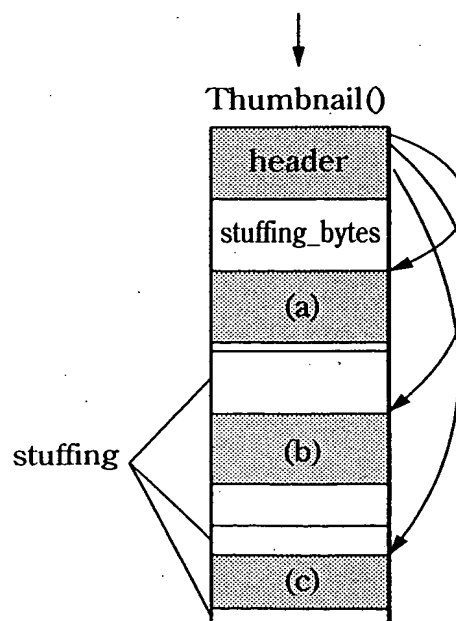
SYNTAX	NUMBER OF BYTES	ABBREVIATION
menu.thmb/mark.thmb() {		
reserved	256	bslbf
Thumbnail()		
for (i=0;i<N1;i++)		
padding_word	16	bslbf
}		

FIG.86

SYNTAX	NUMBER OF BYTES	ABBREVIATION
Thumbnail(){		
<b>version_number</b>	8*4	char
<b>length</b>	32	uimsbf
if (length !=0){		
<b>tn_blocks_start_address</b>	32	bslbf
<b>number_of_thumbnails</b>	16	uimsbf
<b>tn_block_size</b>	16	uimsbf
<b>number_of_tn_blocks</b>	16	uimsbf
reserved	16	bslbf
for (i=0; i<number_of_thumbnails; i++){		
<b>thumbnail_index</b>	16	uimsbf
<b>thumbnail_picture_format</b>	8	bslbf
reserved	8	bslbf
<b>picture_data_size</b>	32	uimsbf
<b>start_tn_block_number</b>	16	uimsbf
<b>x_picture_length</b>	16	uimsbf
<b>y_picture_length</b>	16	uimsbf
reserved	16	uimsbf
}		
<b>stuffing_bytes</b>	8*2*L1	bslbf
for(k=0; k<number_of_tn_blocks; k++){		
<b>tn_block</b>	tn_block_size*1024*8	
}		
}		

FIG.87

Thumbnail_picture_format	MEANING
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

**FIG.88****FIG.89B**

## DVR MPEG-2 TRANSPORT STREAM

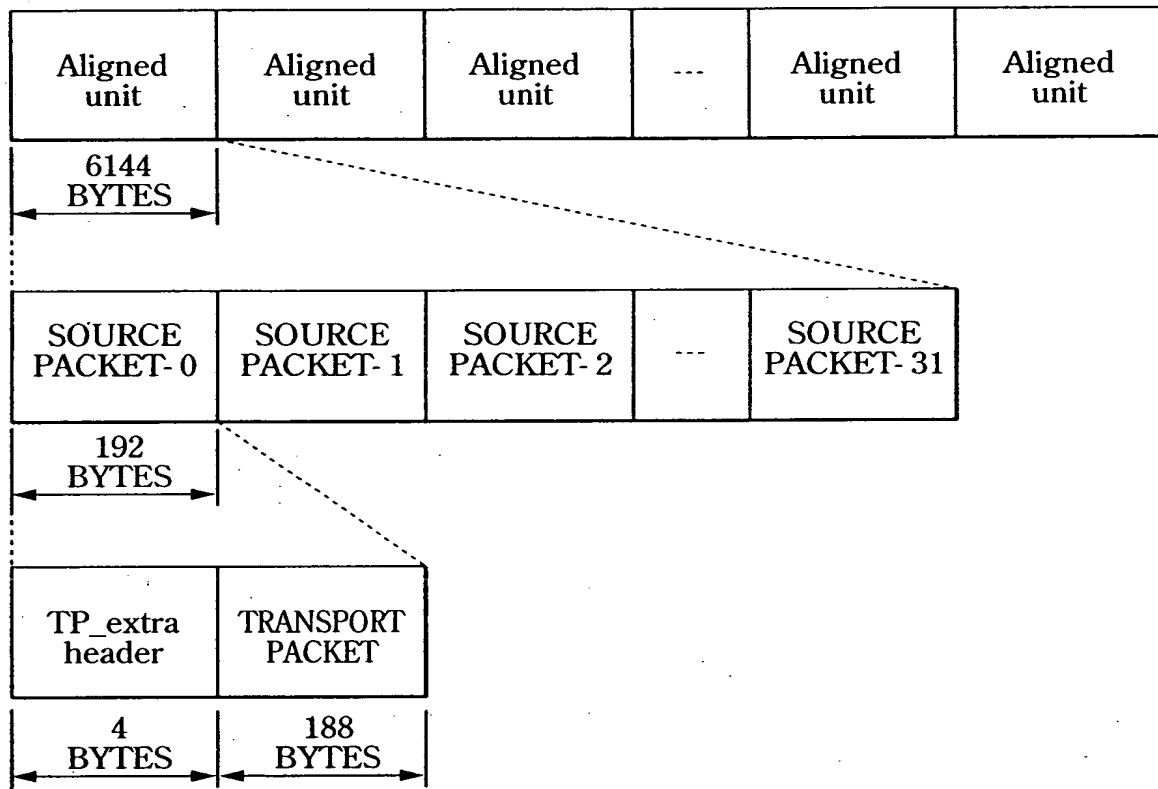
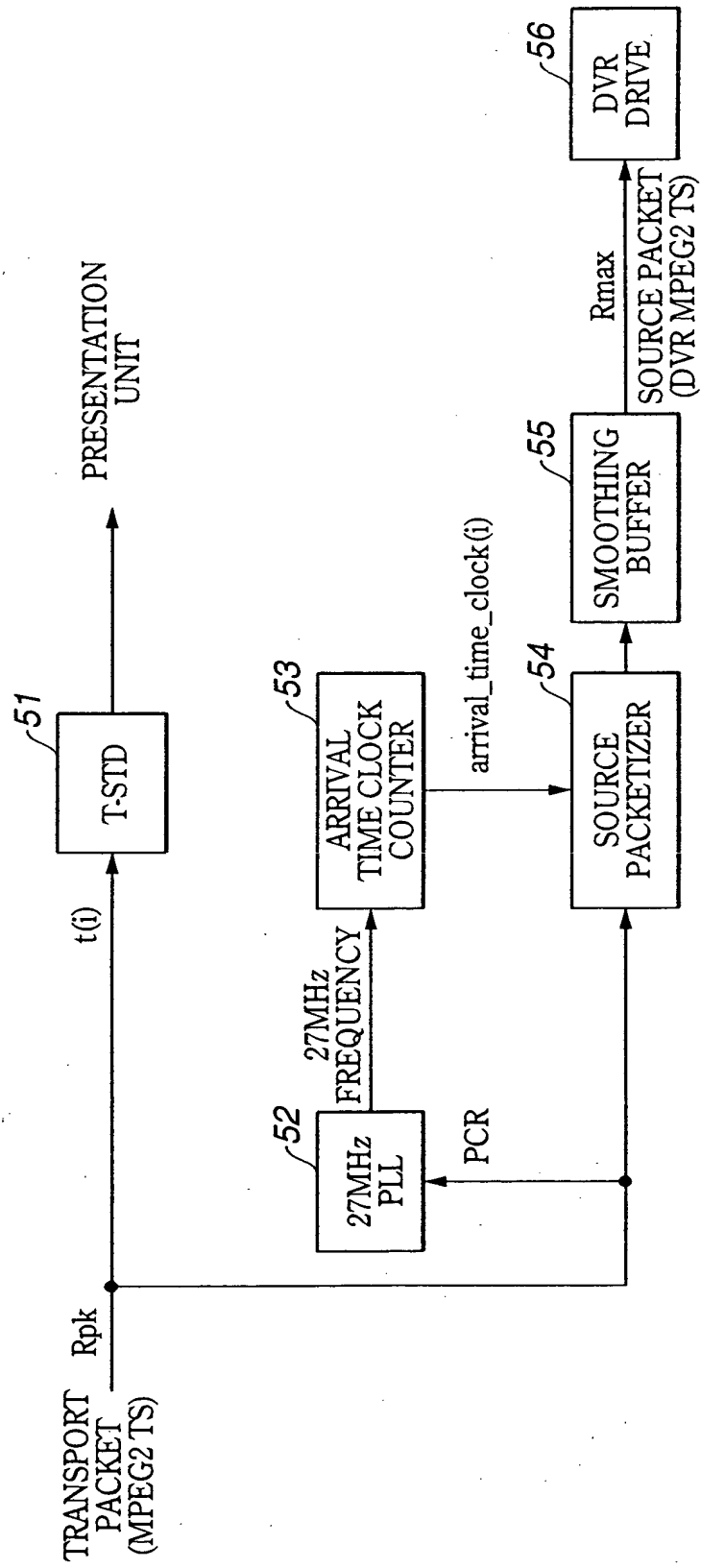


FIG.90

**FIG.91**

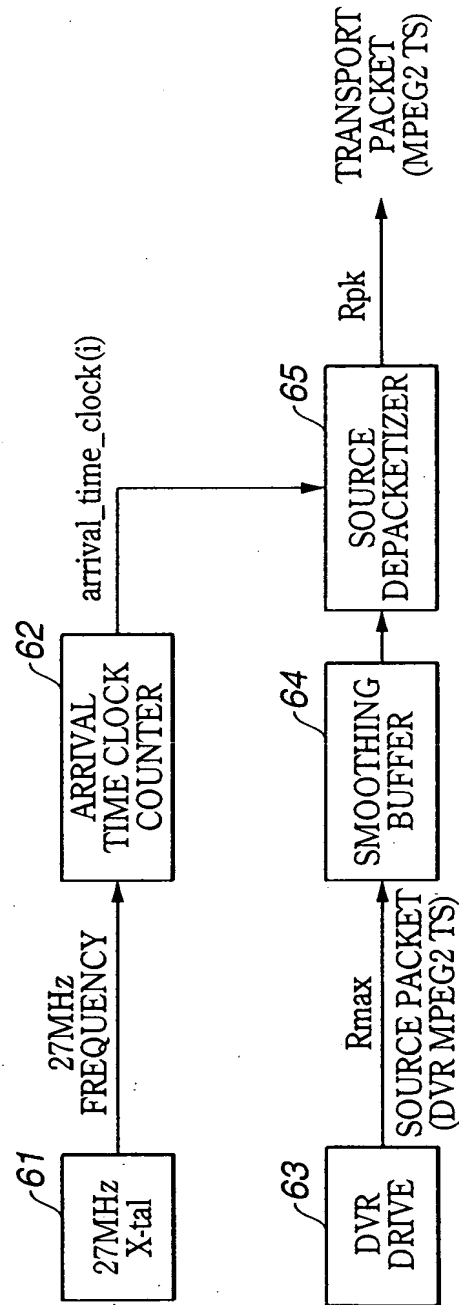


FIG.92

SYNTAX	NUMBER OF BYTES	ABBREVIATION
source_packet() {		
TP_extra_header()		
trasport_packet()		
}		

FIG.93



SYNTAX	NUMBER OF BYTES	ABBREVIATION
TP_extra_header() {		
copy_permission_indicator	2	uimsbf
arrival_time_stamp	30	uimsbf
}		

FIG.94

copy_permission _indicator	MEANING
00	copy free
01	no more copy
10	copy once
11	copy prohibited

FIG.95

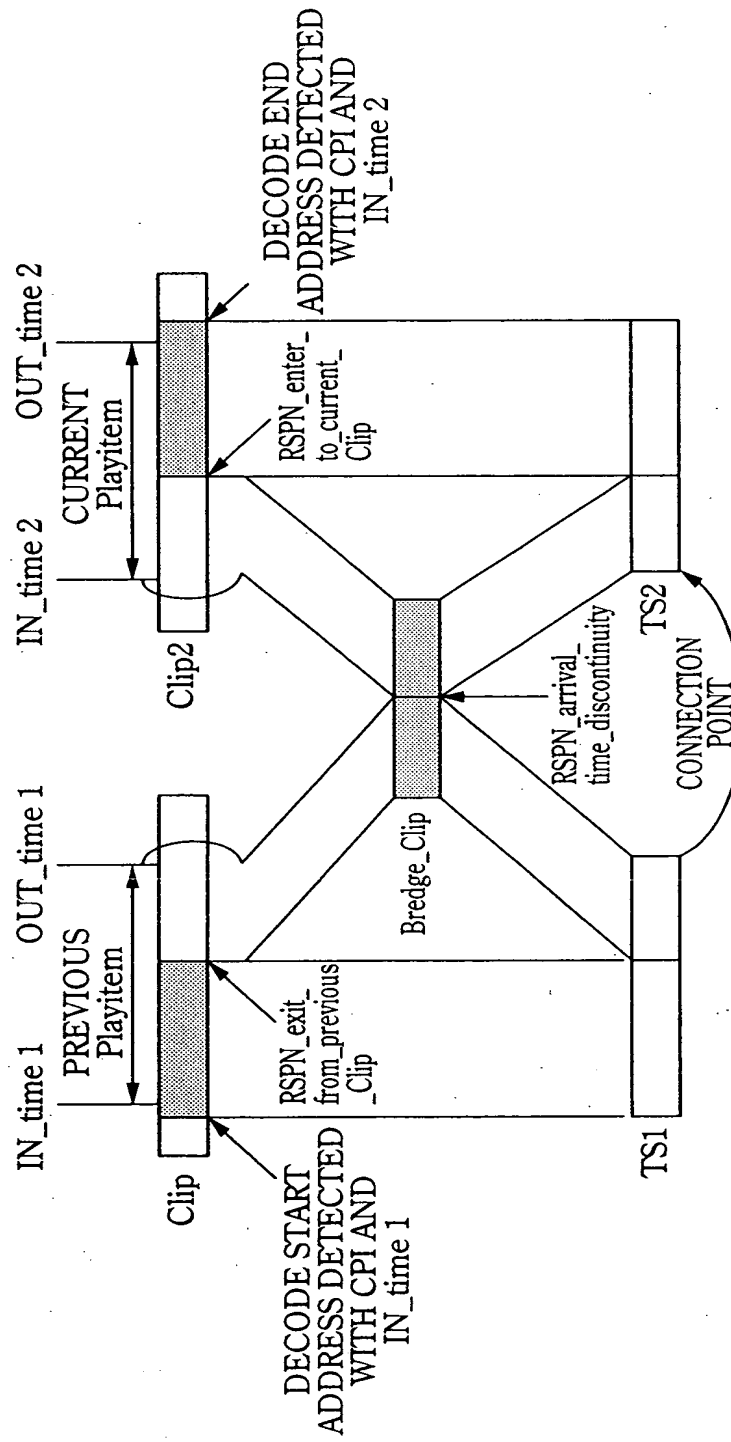


FIG.96

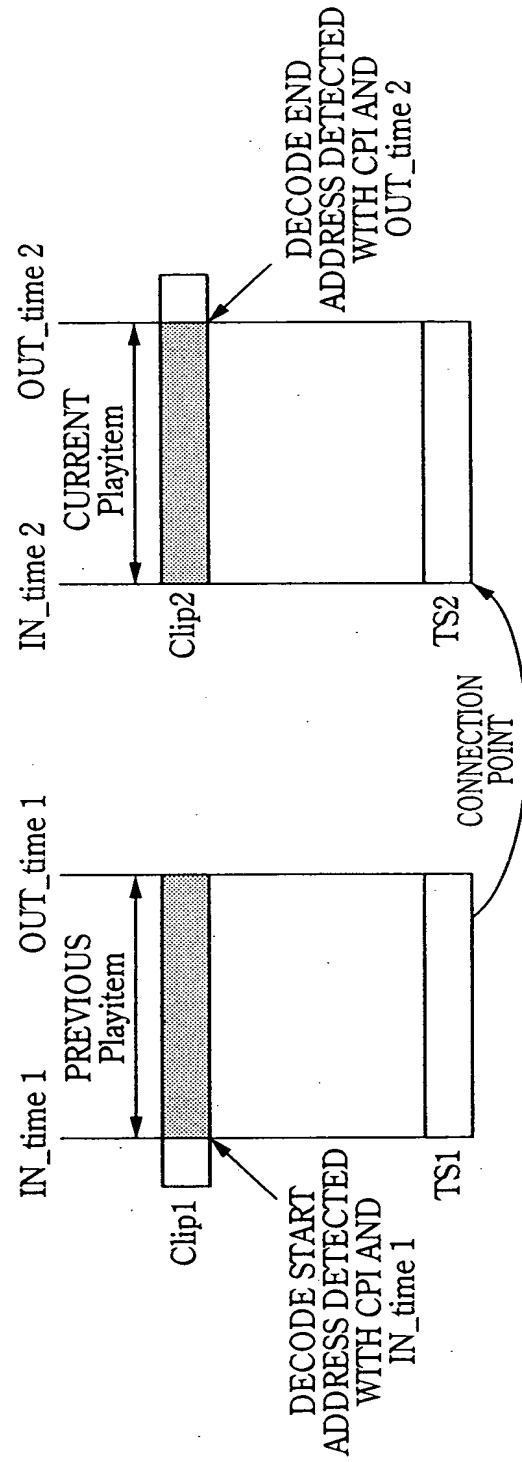
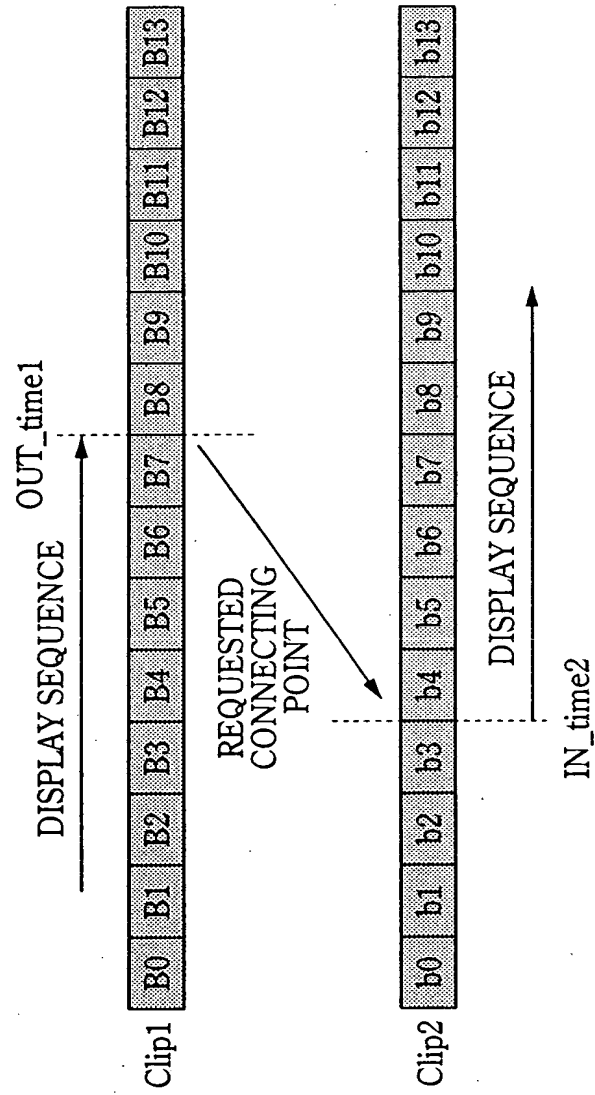


FIG.97

**FIG.98**

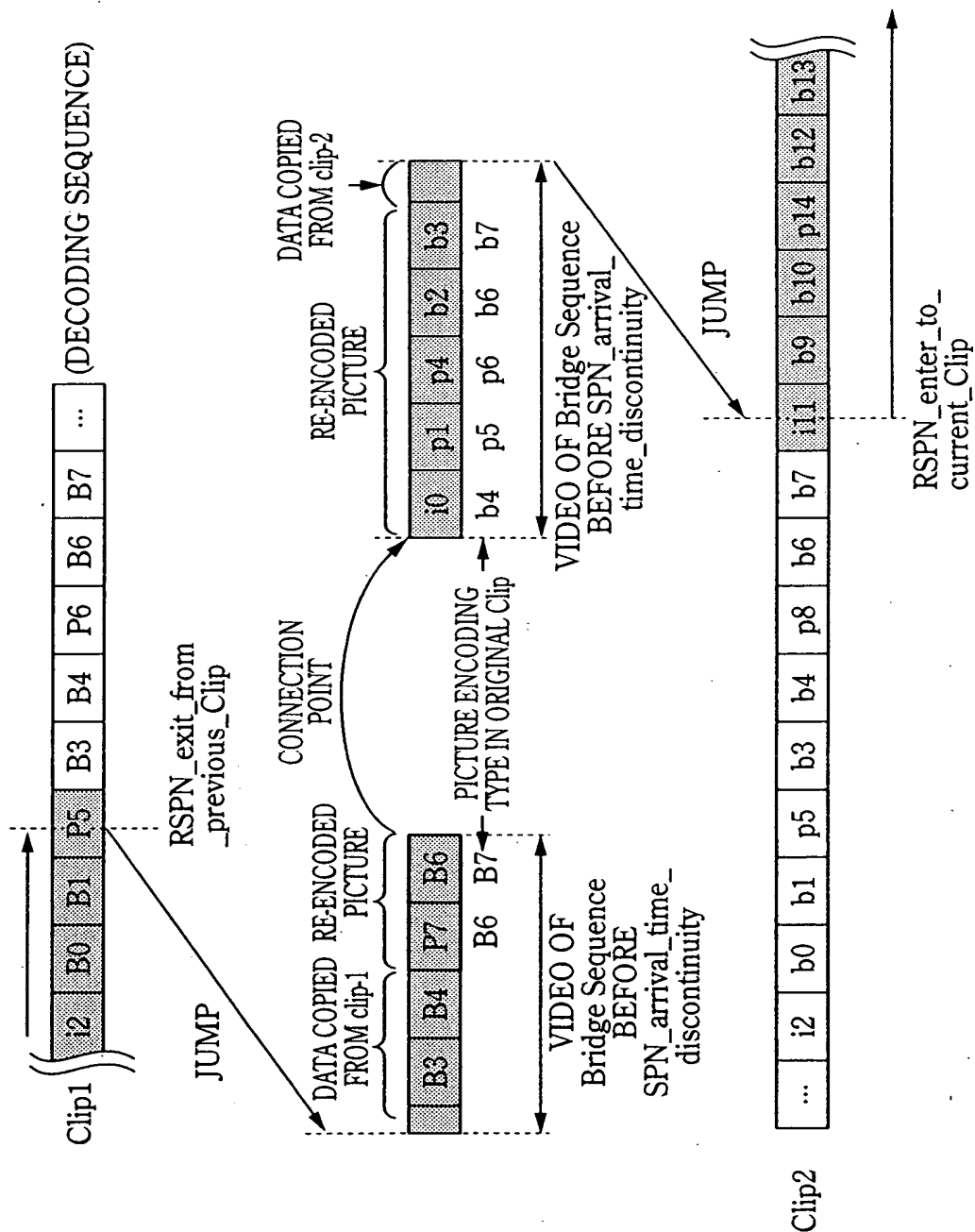
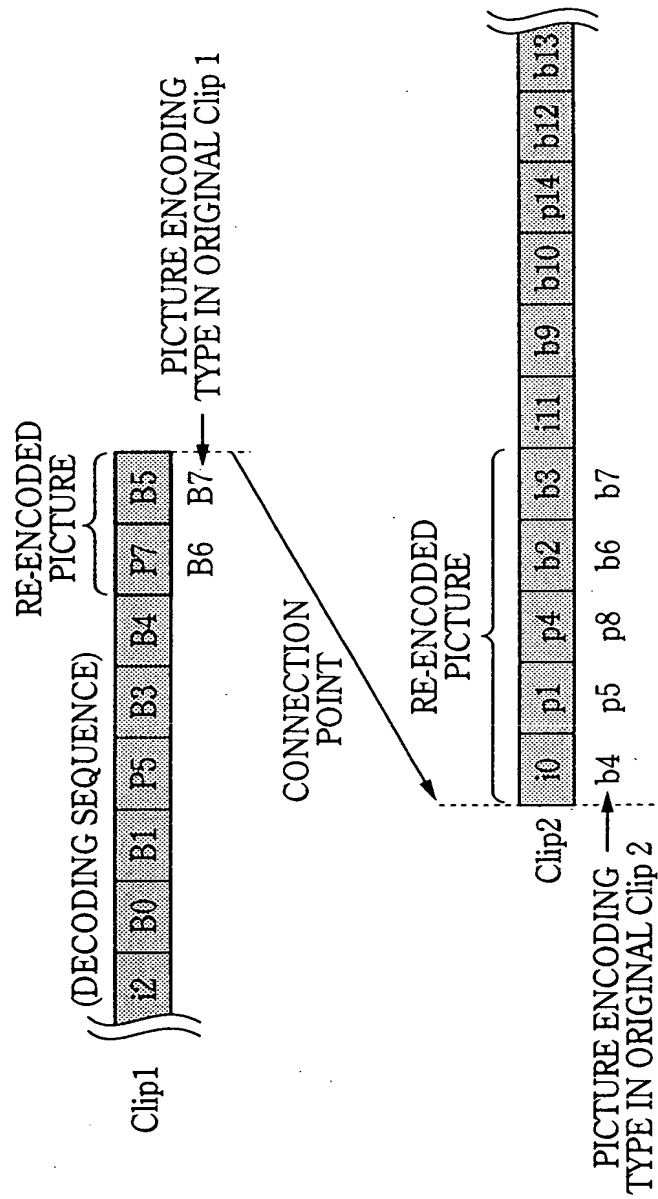
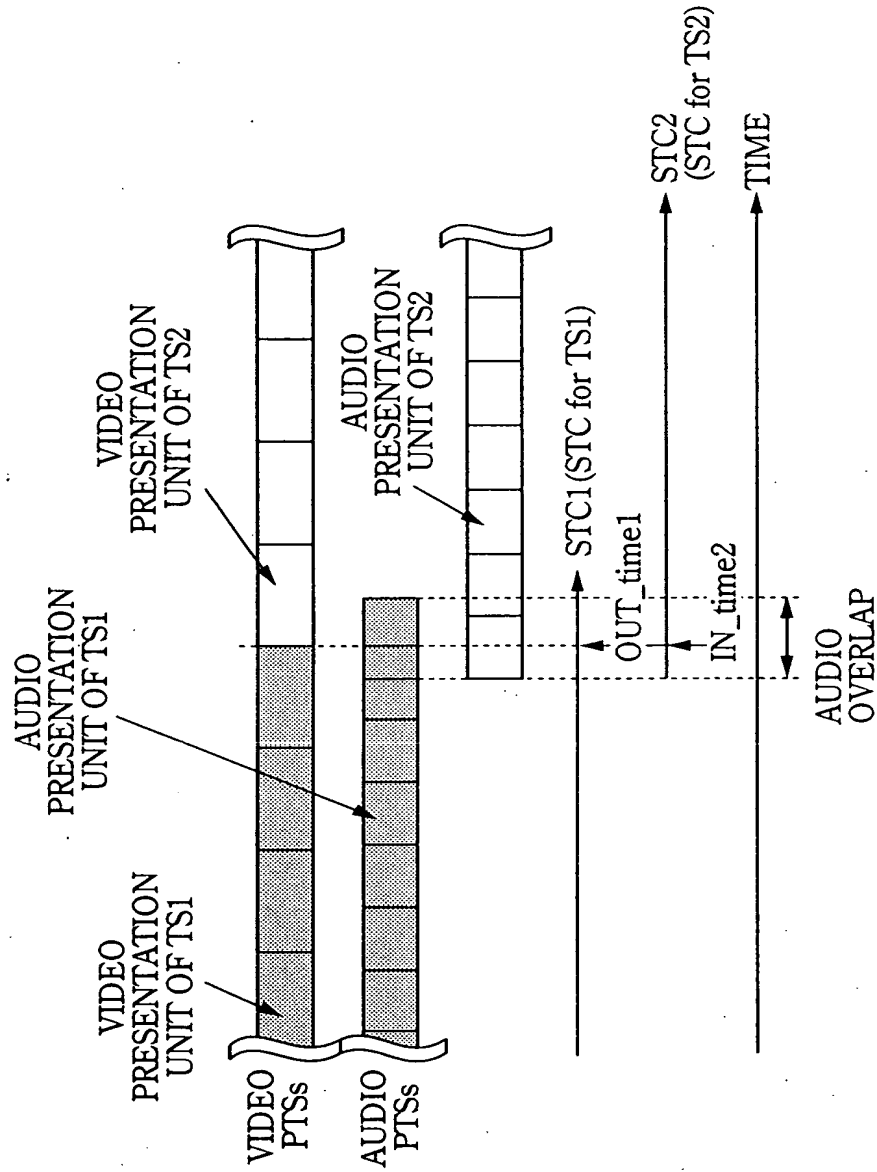
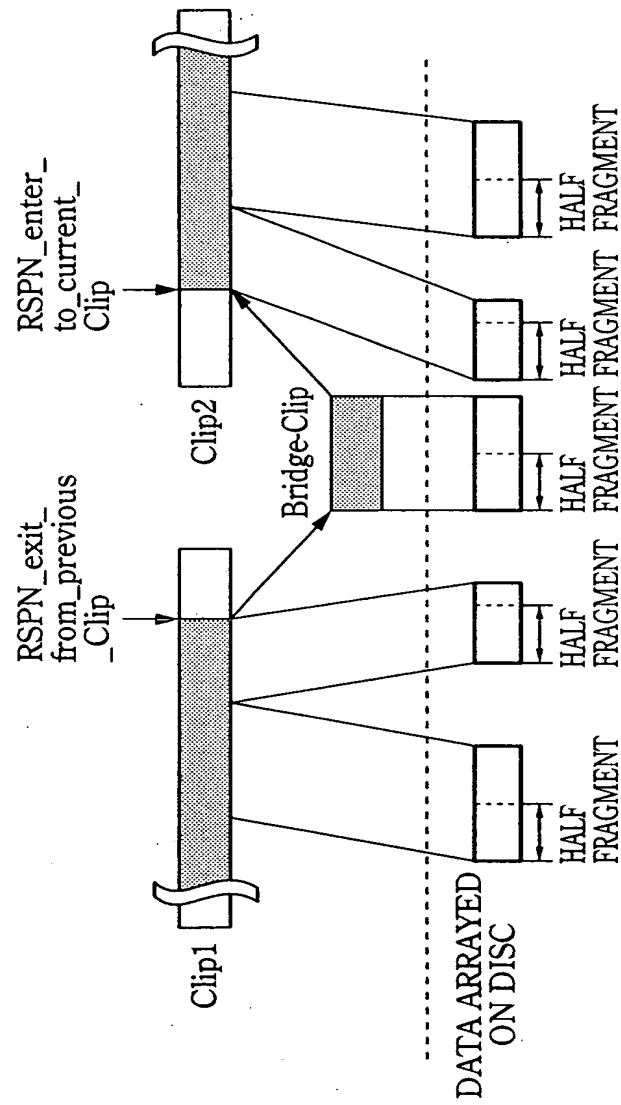


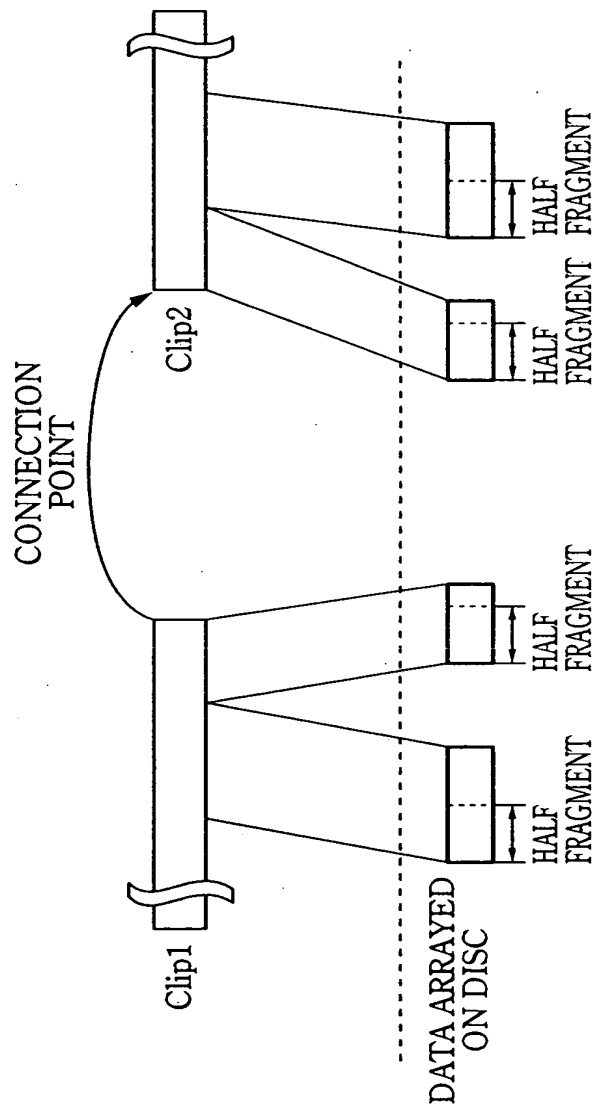
FIG.99

**FIG.100**

**FIG.101**



**FIG.102**



**FIG.103**

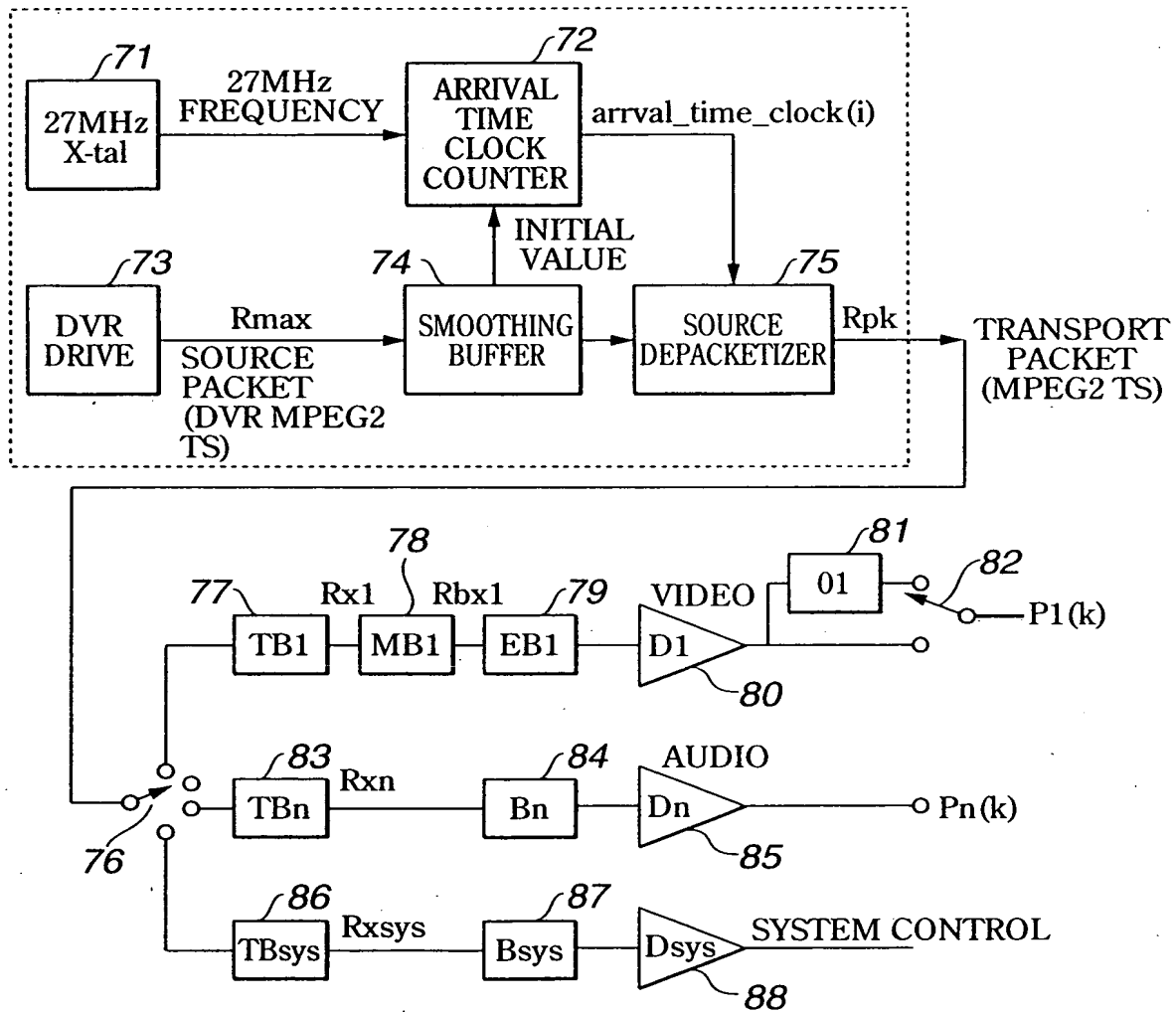


FIG. 104

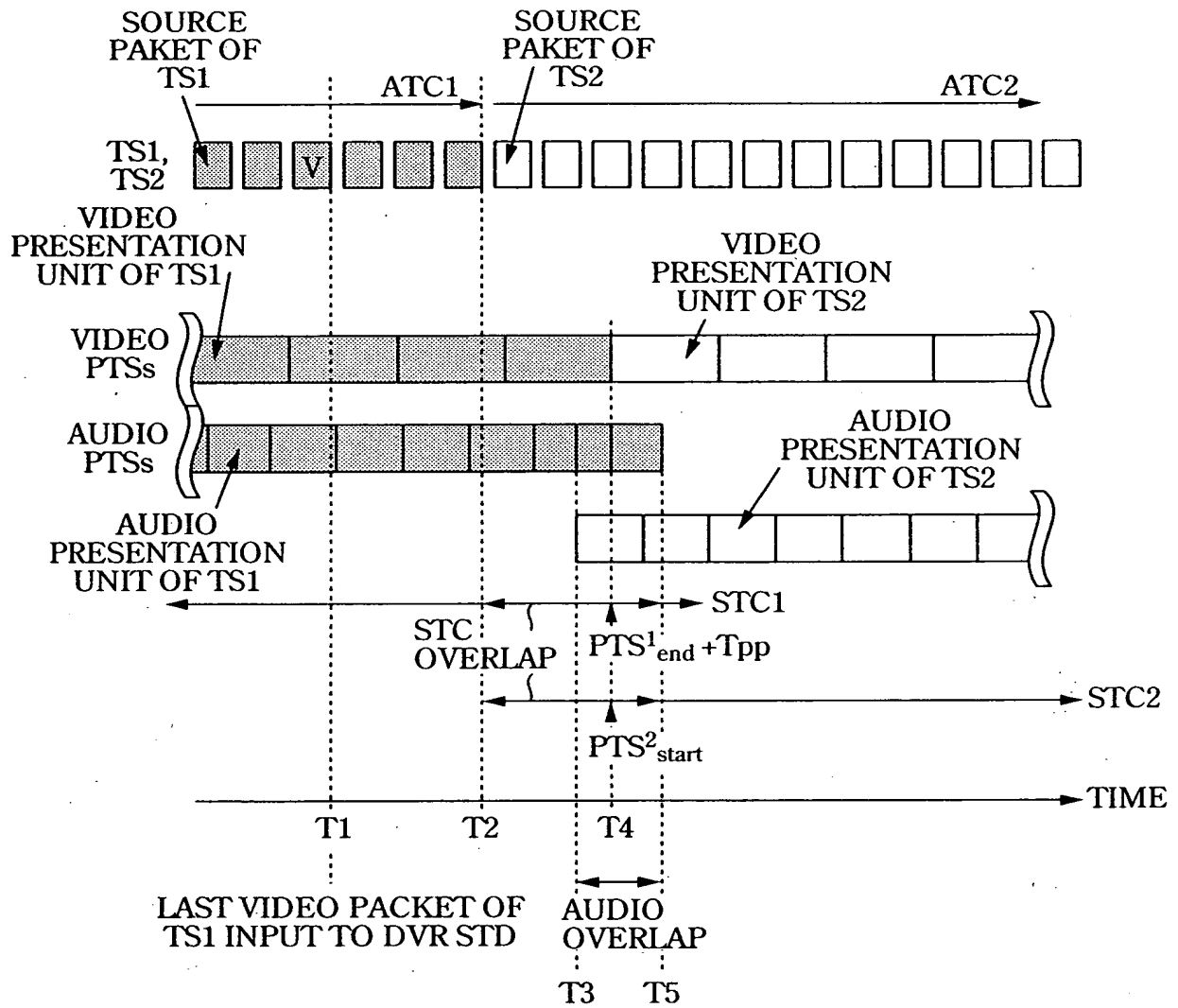


FIG.105

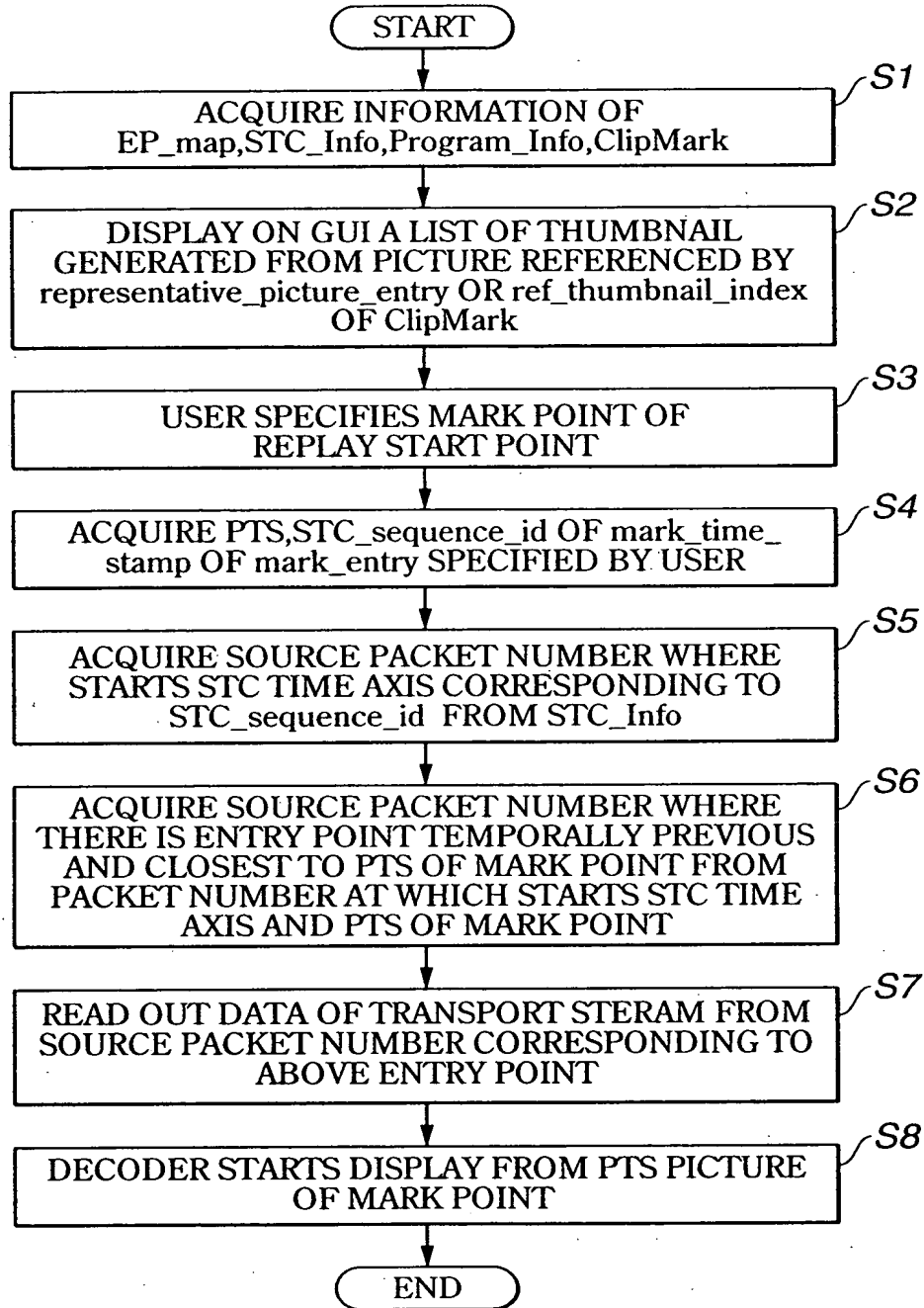


FIG.106

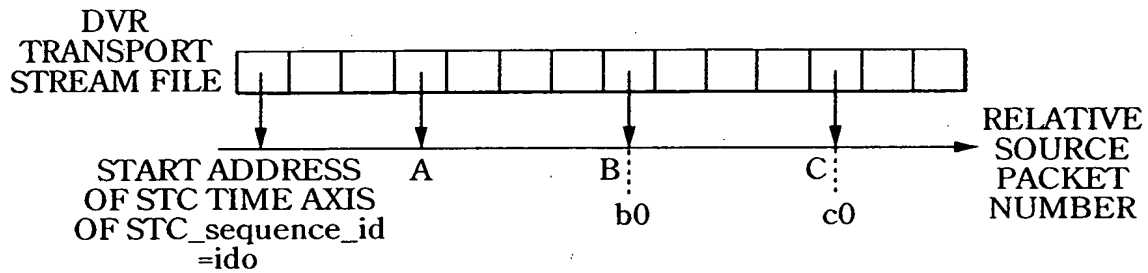


FIG.107

EP\_map

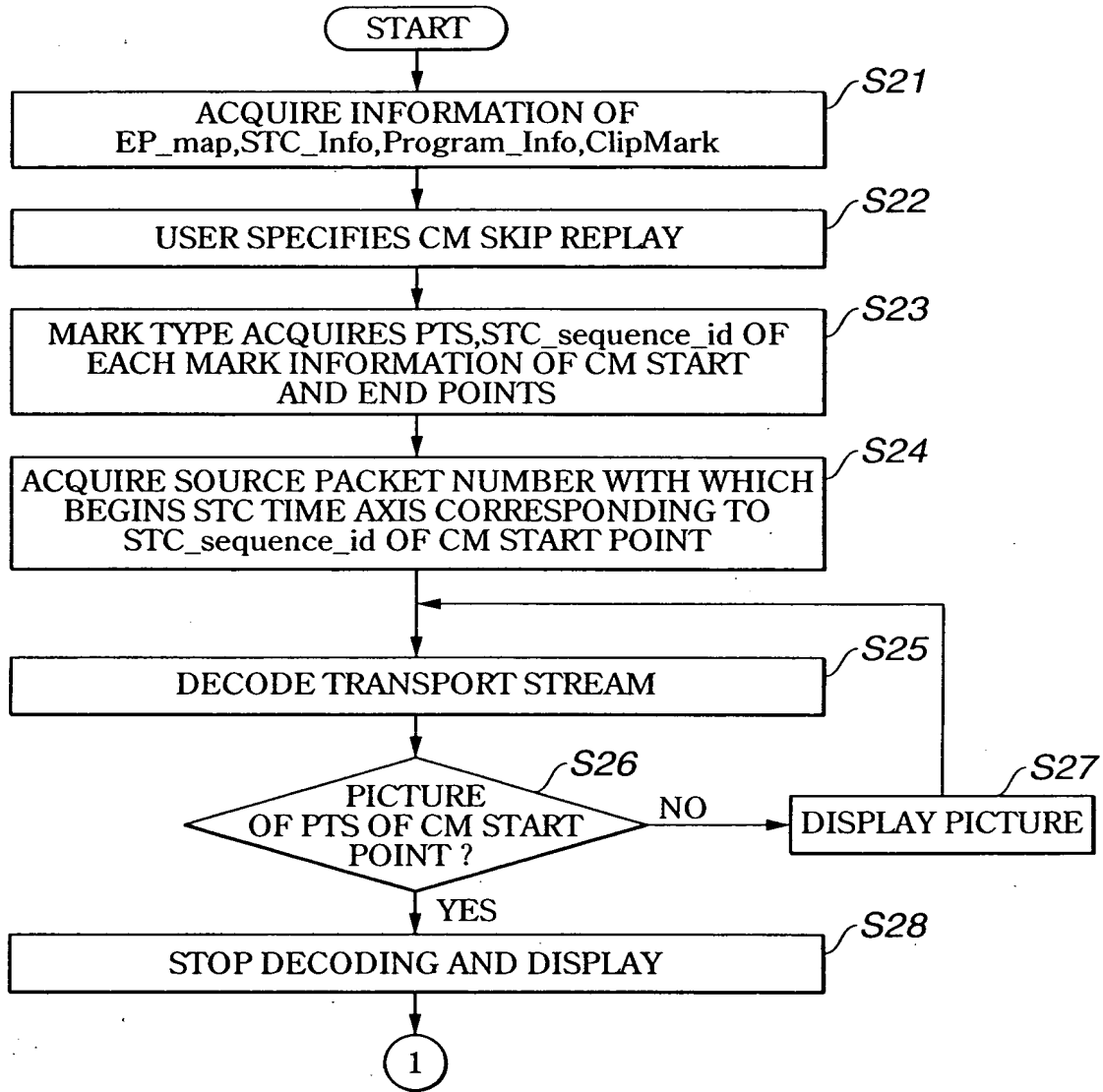
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

FIG.108

ClipMark

Mark_type	mark_entry		representative_picture_entry	
	Mark_Time_stamp	STC_sequence_id	Mark_Time_stamp	STC_sequence_id
...	...	...	...	...
0x92(scene start)	PTS(a1)	id0	PTS(a2)	id0
0x94(CM start)	PTS(b0)	id0	PTS(b0)	id0
0x95(CM end)	PTS(c0)	id0	PTS(c0)	id0
...	...	...	...	...

FIG.109

**FIG.110**

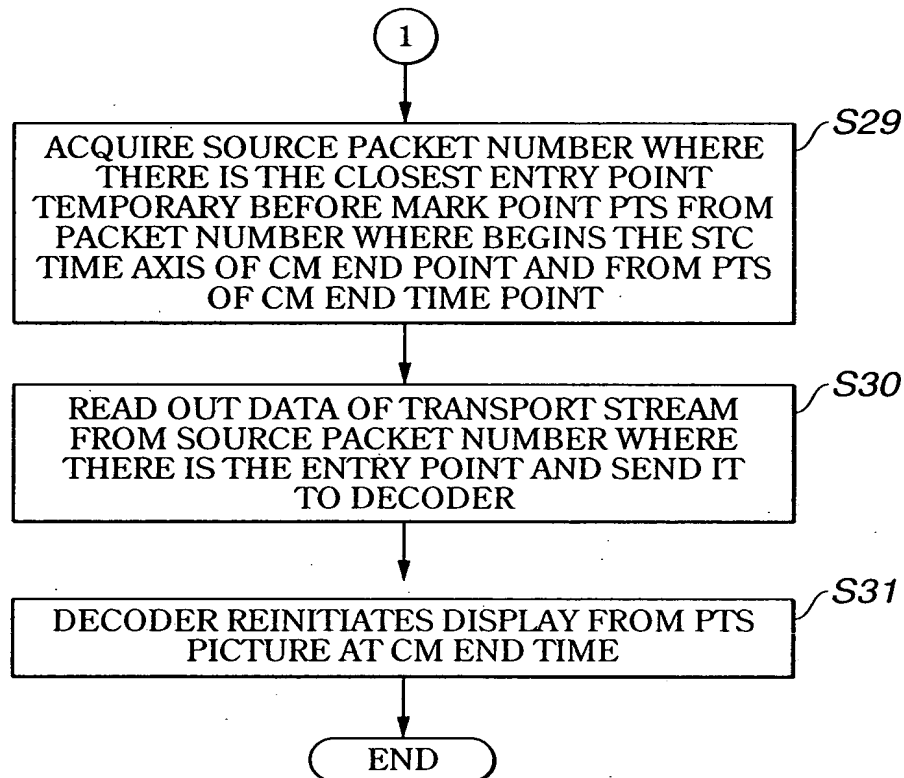


FIG.111



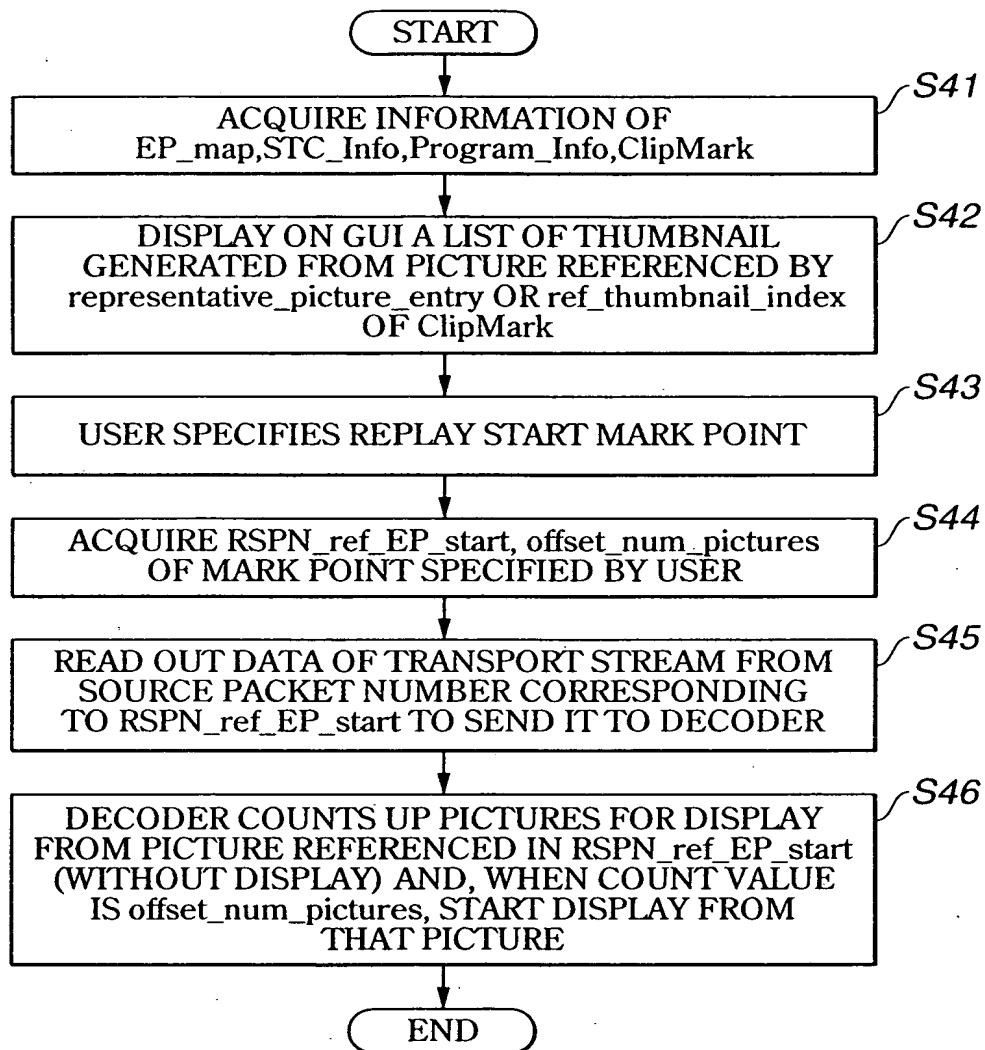


FIG.112

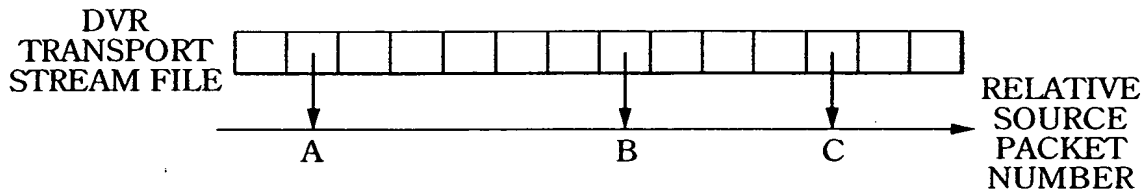


FIG.113

EP\_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

FIG.114

ClipMark

mark_type	mark_entry		representative_picture_entry	
	RSPN_ref_EP_start	offset_num_pictures	RSPN_ref_EP_start	offset_num_pictures
...	...	...	...	...
0x92(scene start)	A	M1	A	M2
0x94(CM start)	B	N1	B	N1
0x95(CM end)	C	N2	C	N2
...	...	...	...	...

FIG.115

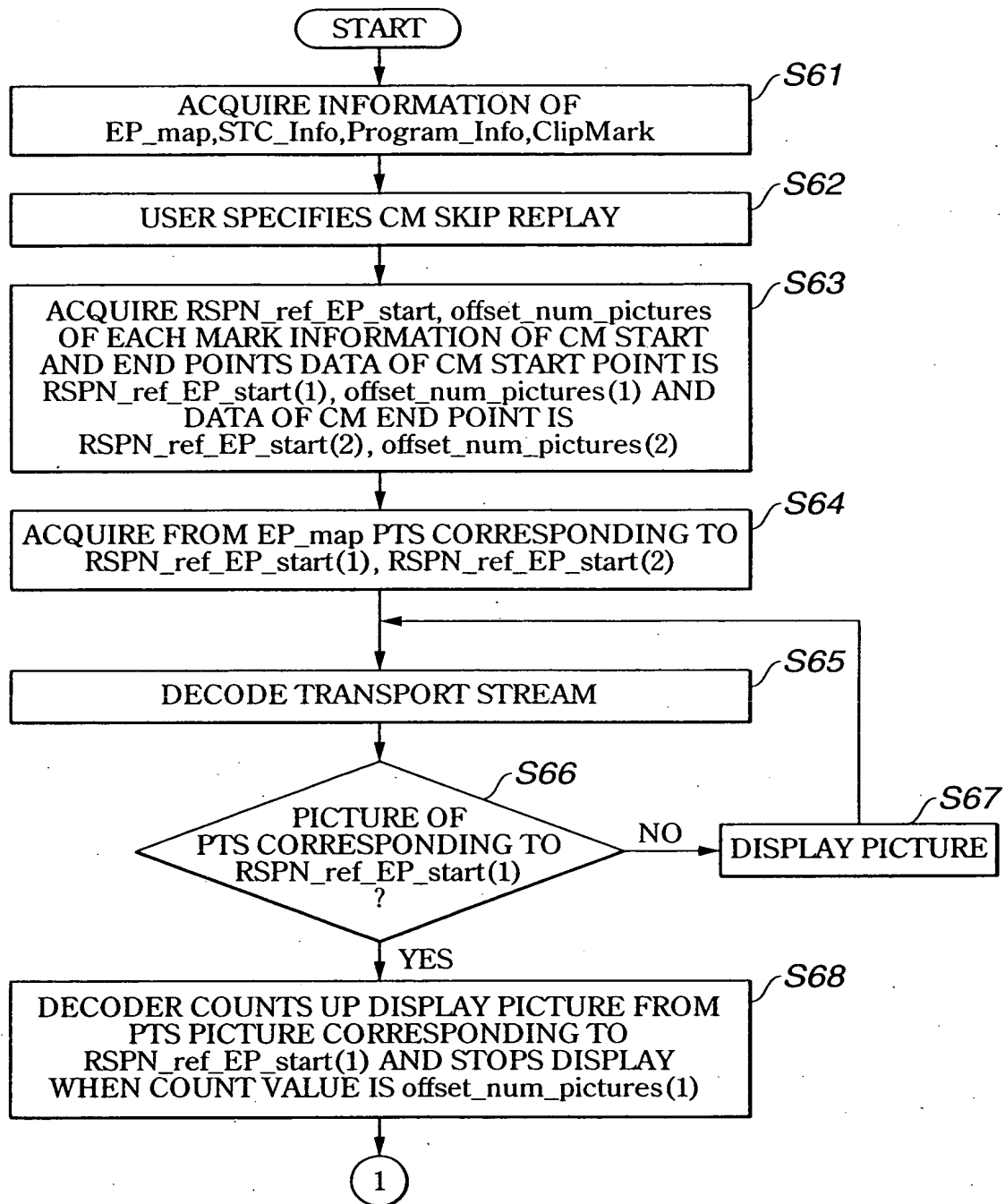


FIG.116

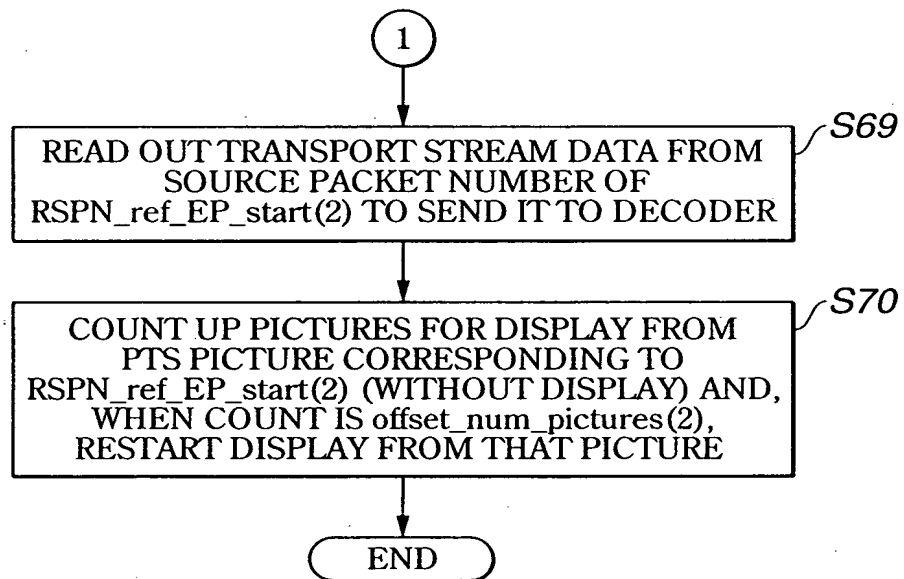
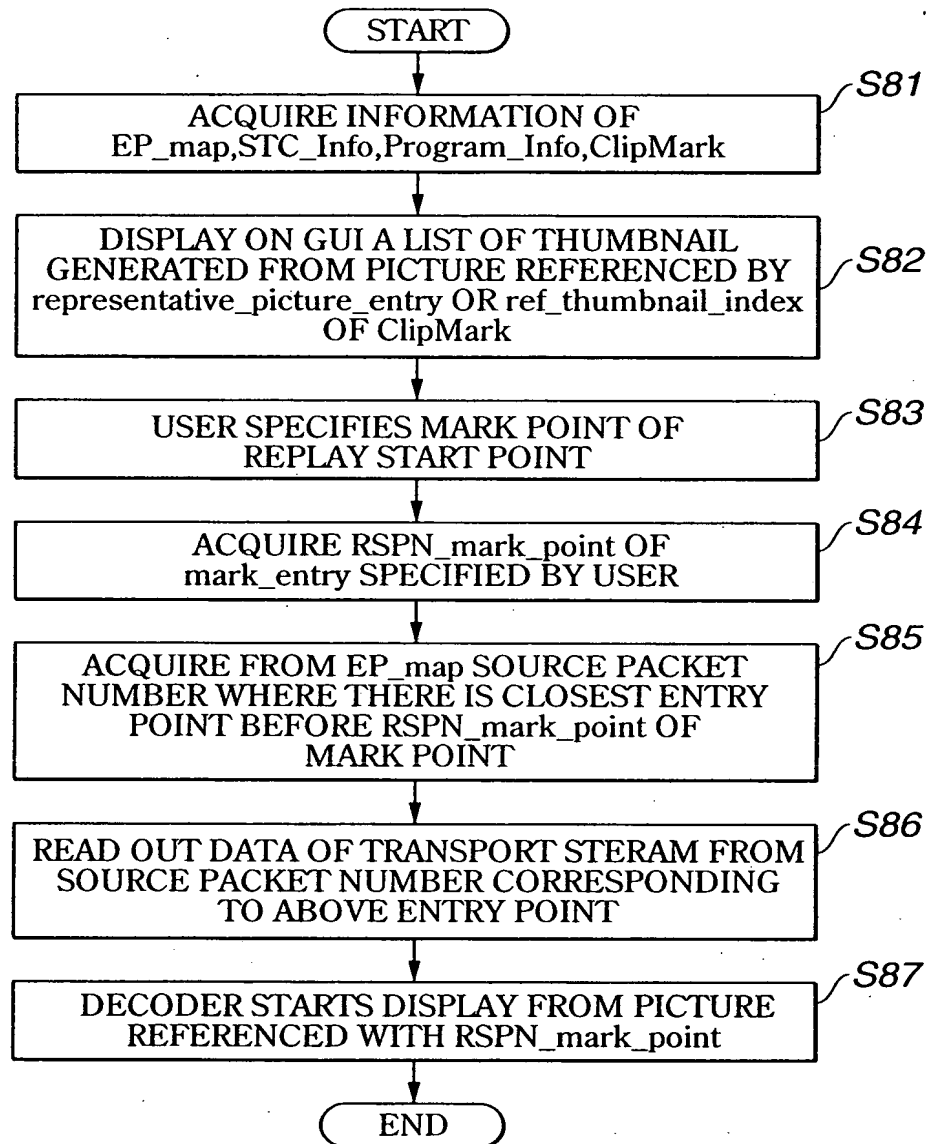


FIG.117

**FIG.118**

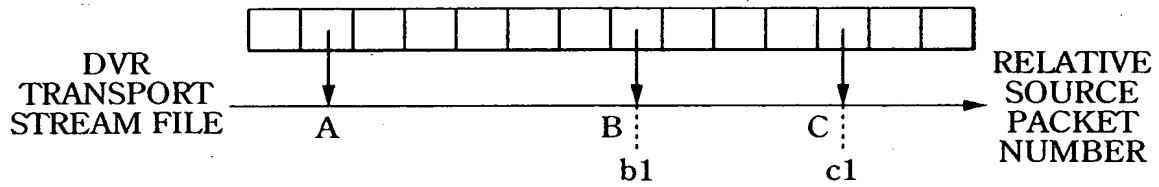


FIG.119

EP\_map

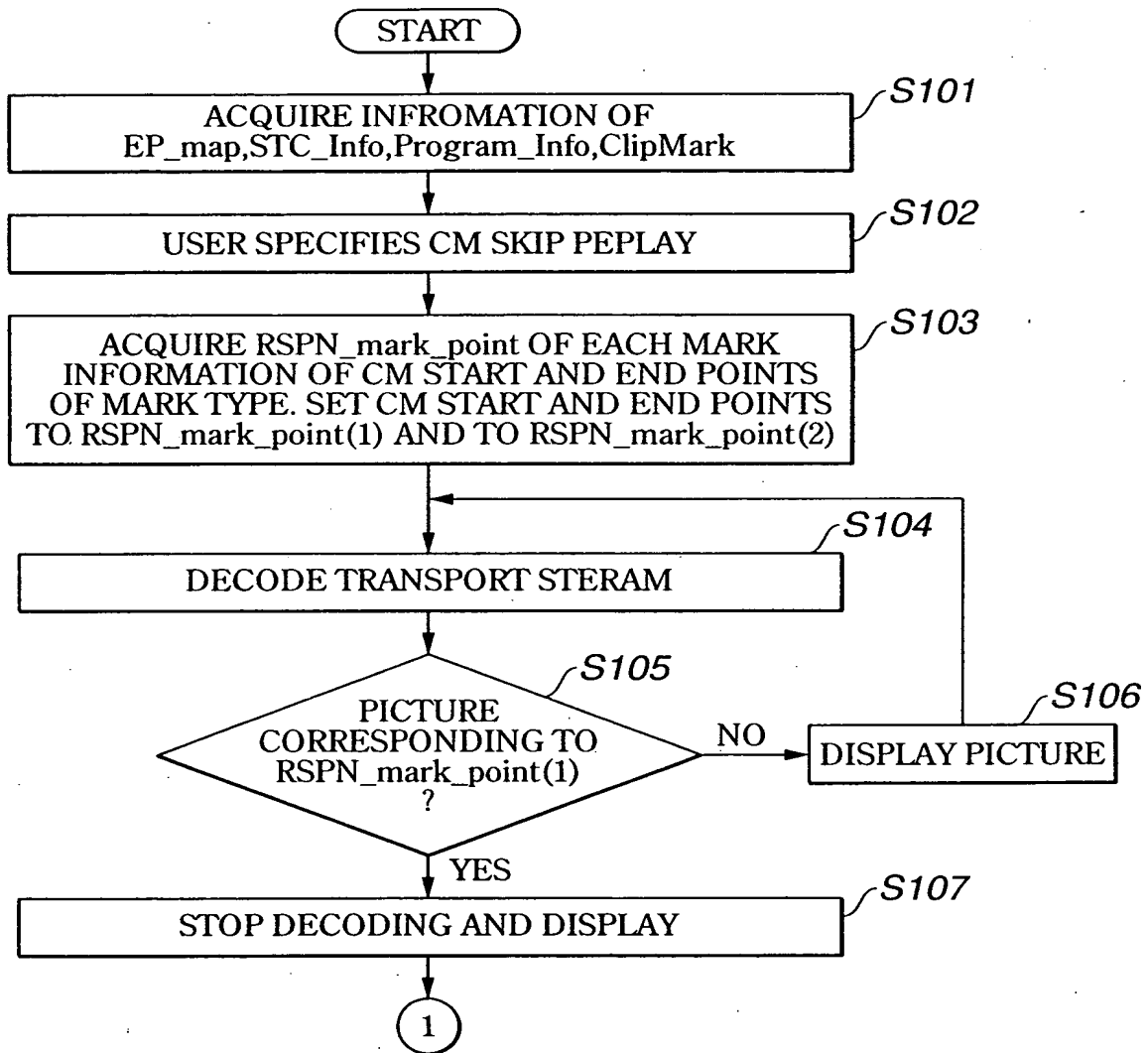
RSPN_EP_start	PTS_EP_start
...	...
A	PTS(A)
B	PTS(B)
C	PTS(C)
...	...

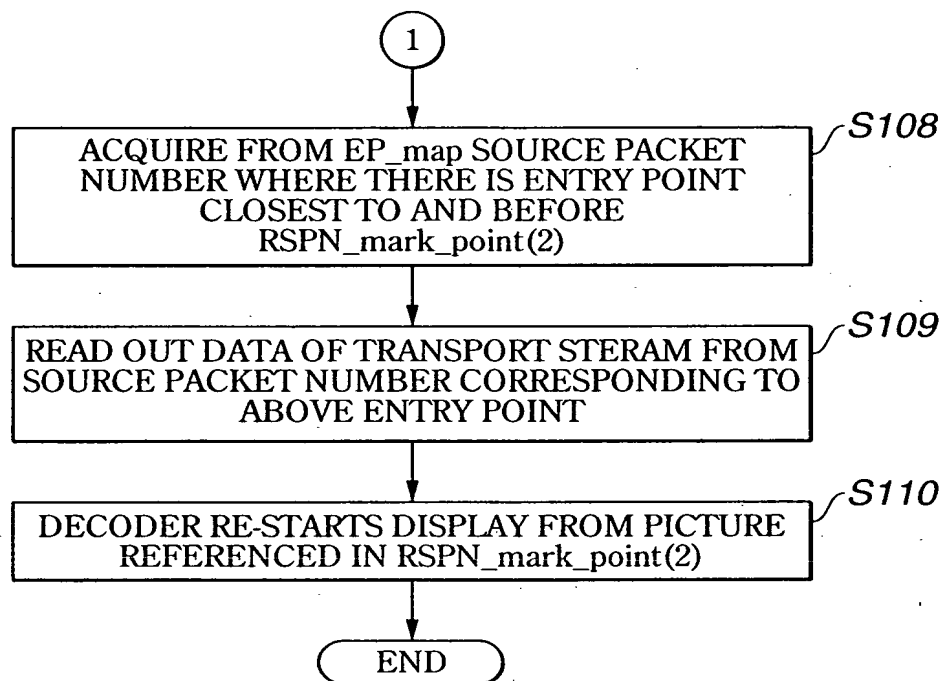
FIG.120

ClipMark

mark_type	mark_entry	representative_picture_entry
	RSPN_mark_point	RSPN_mark_point
...	...	...
0x92(scene start)	a1	a2
0x94(CM start)	b1	b1
0x95(CM end)	c1	c1
...	...	...

FIG.121

**FIG.122**

**FIG.123**



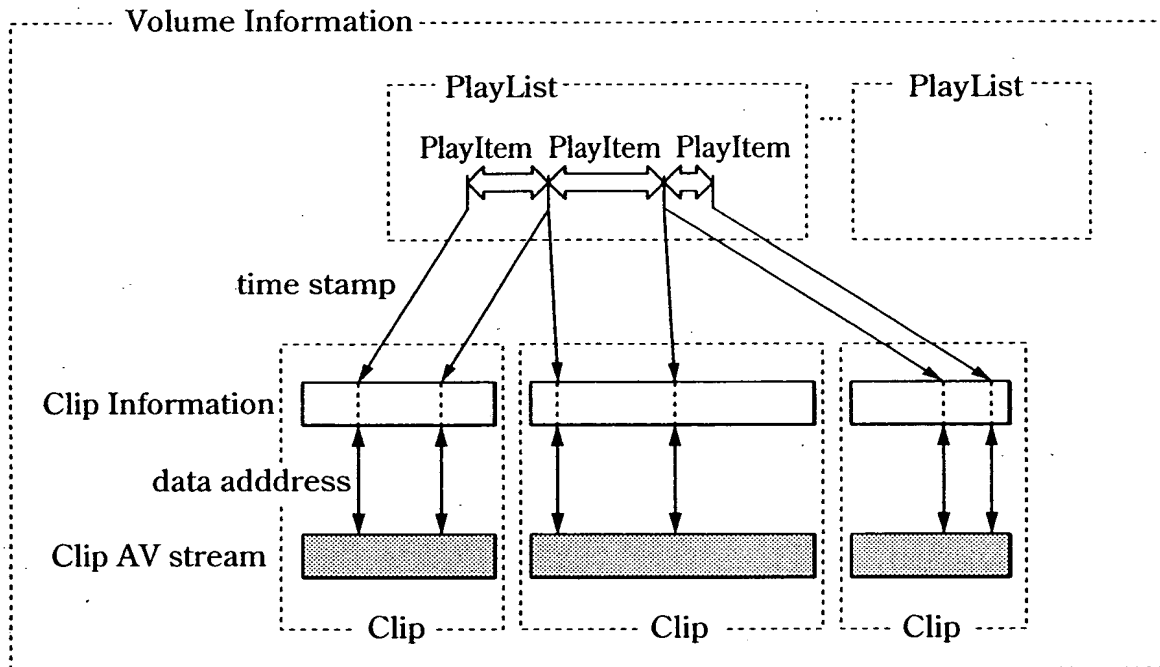


FIG.124

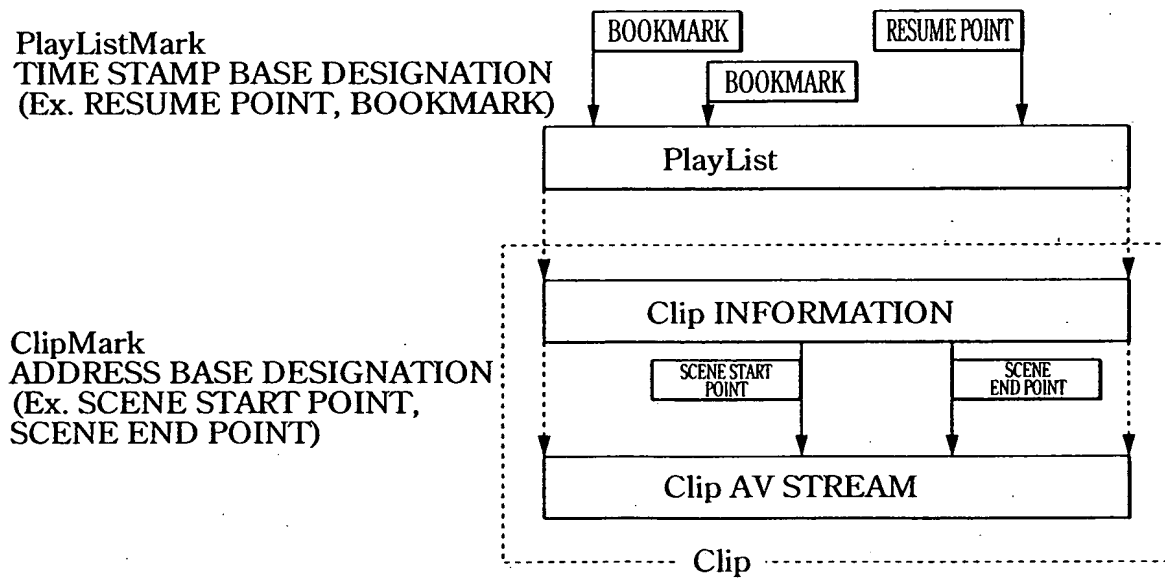


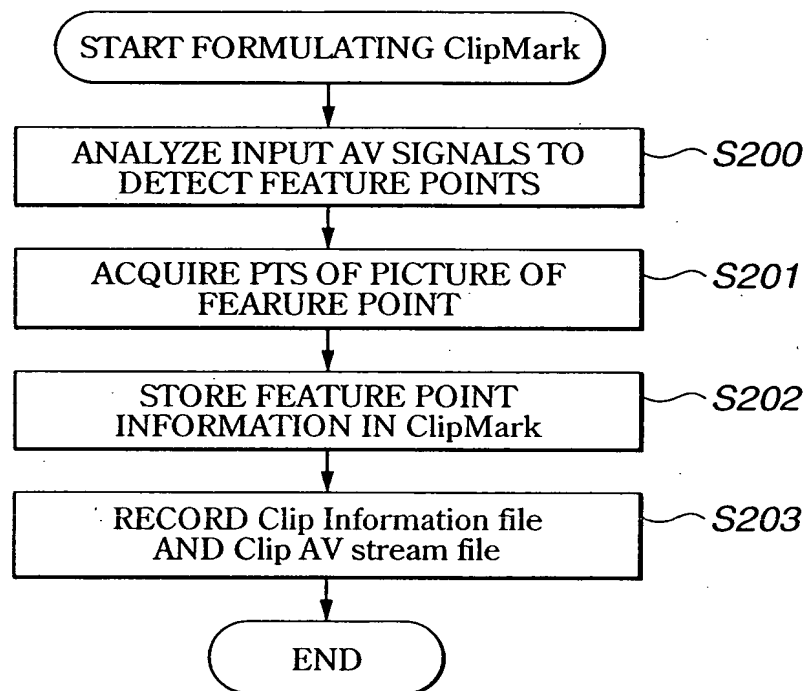
FIG.125

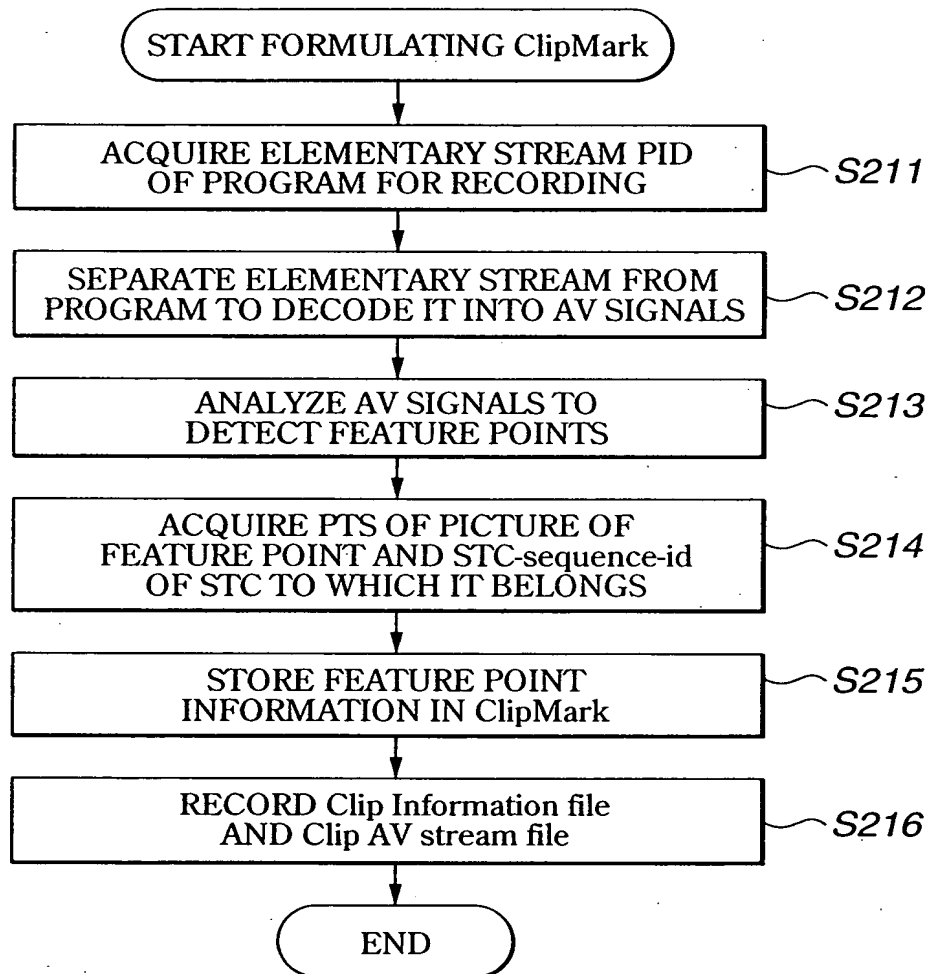
SYNTAX	NUMBER OF BYTES	ABBREVIATION
ClipMark(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_Clip_marks</b>	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bslbf
<b>mark_type</b>	8	bslbf
<b>RSPN_mark</b>	32	uimsbf
reserved	32	bslbf
<b>ref_thumbnail_index</b>	16	uimsbf
}		
}		

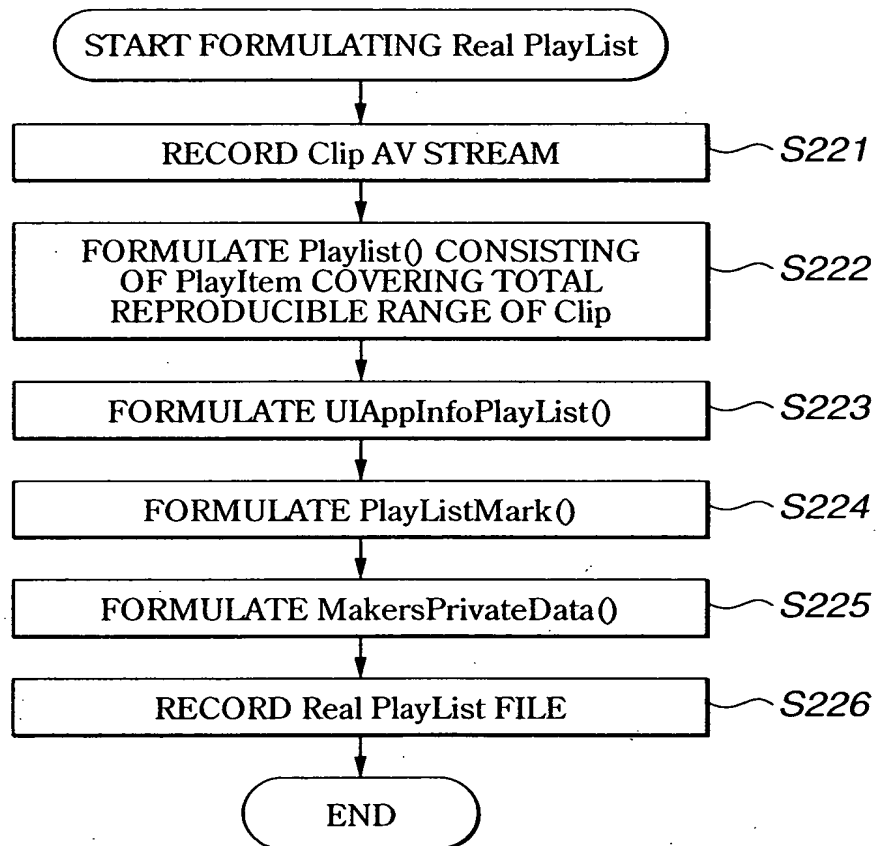
FIG.126

SYNTAX	NUMBER OF BYTES	ABBREVIATION
ClipMark(){		
<b>version_number</b>	8*4	bslbf
<b>length</b>	32	uimsbf
<b>number_of_Clip_marks</b>	16	uimsbf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bslbf
<b>mark_type</b>	8	bslbf
<b>RSPN_ref_EP_start</b>	32	uimsbf
<b>offset_num_pictures</b>	32	uimsbf
<b>ref_thumbnail_index</b>	16	uimsbf
}		
}		

FIG.127

**FIG.128**

**FIG.129**

**FIG.130**

110/118

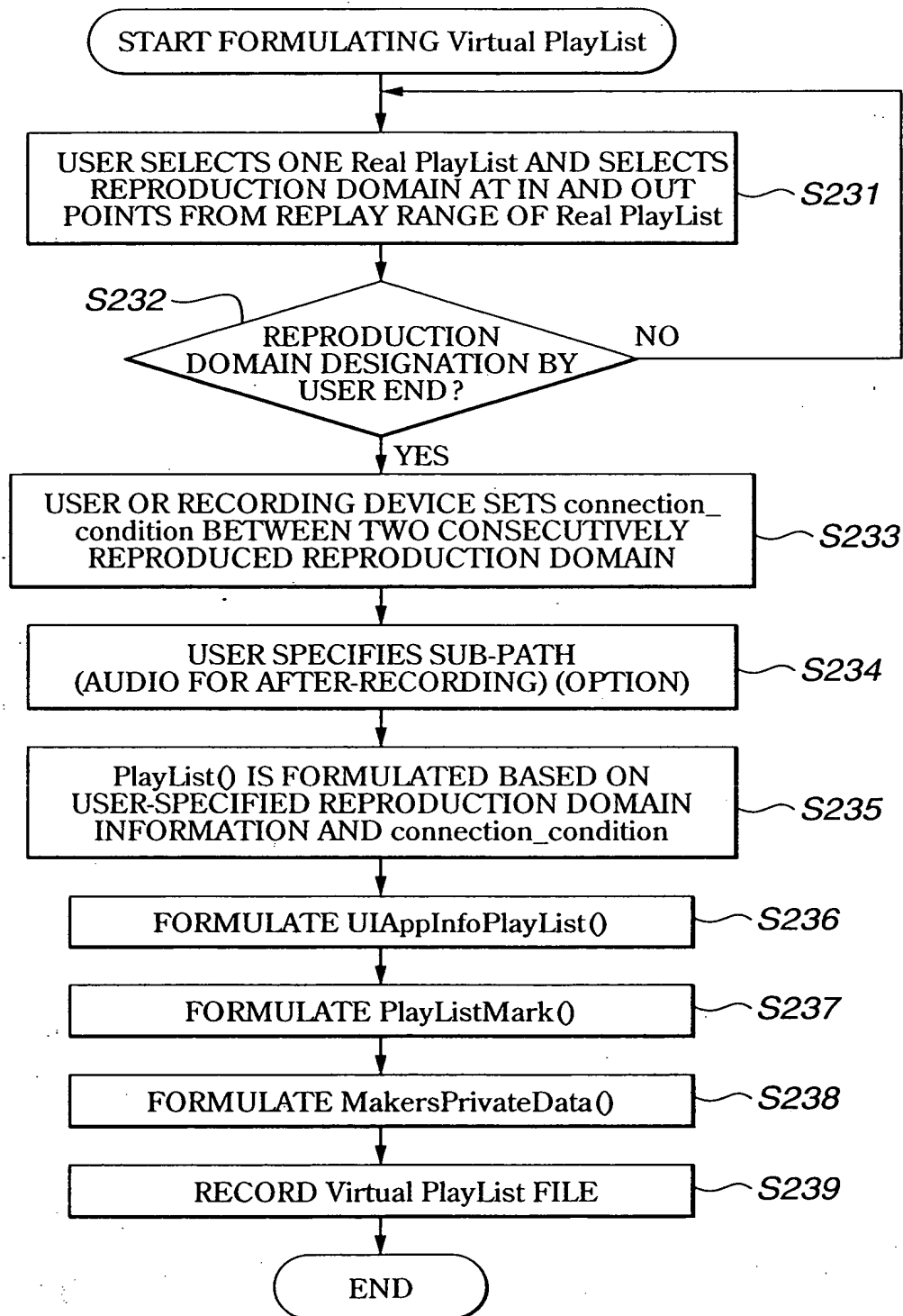


FIG.131

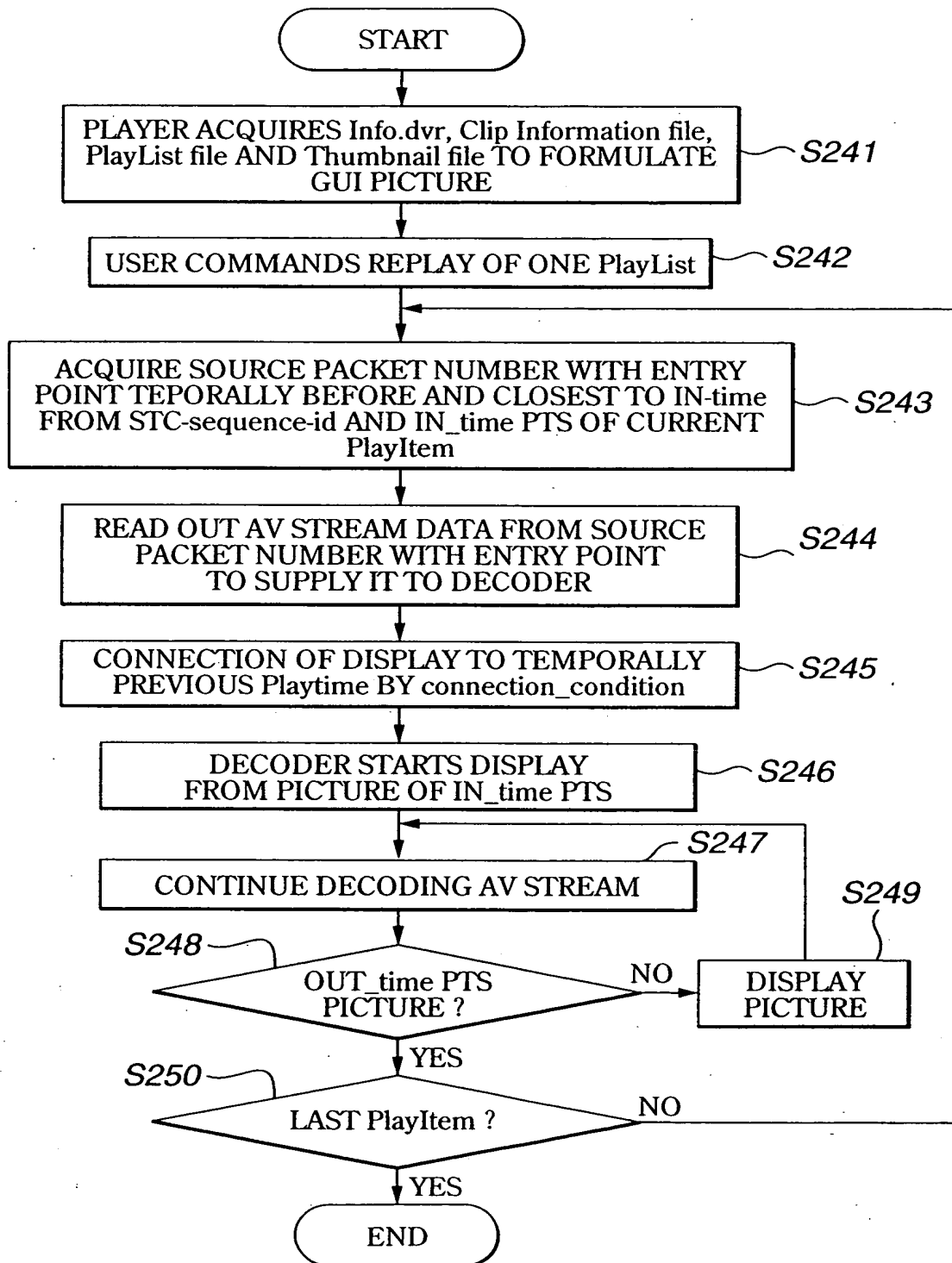
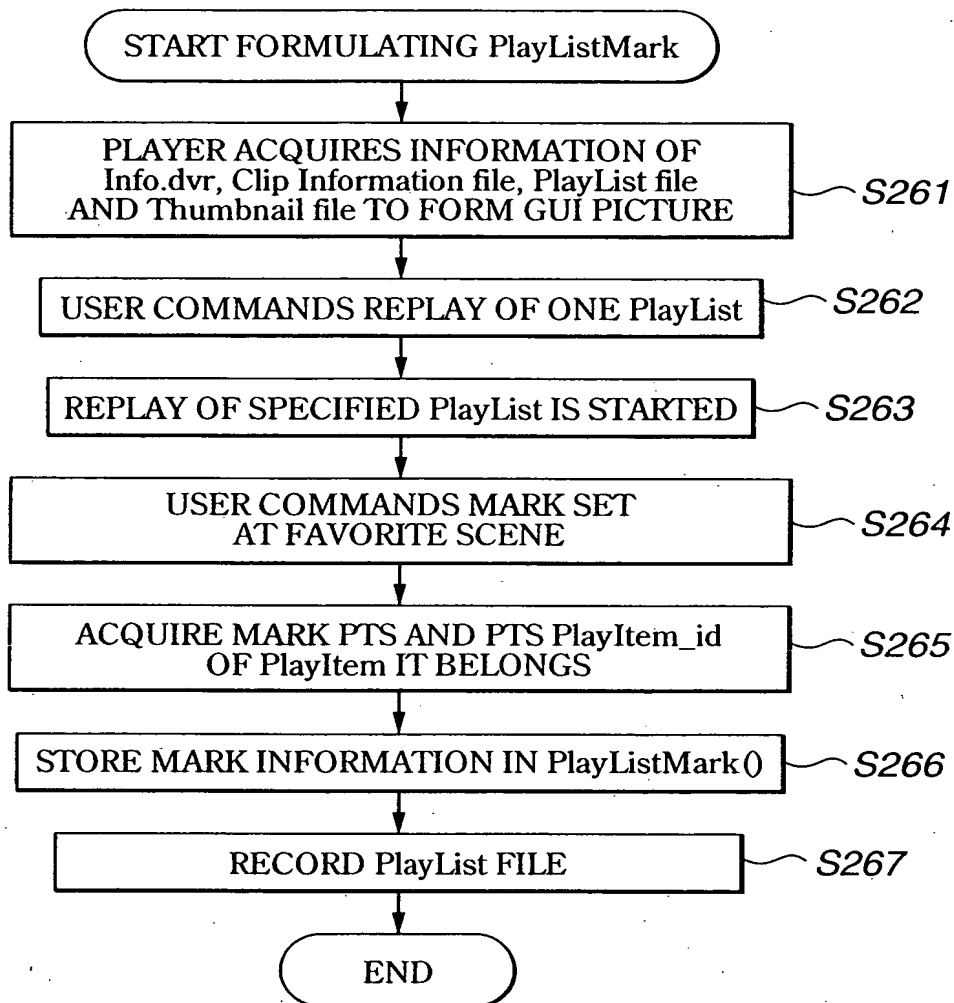


FIG.132

**FIG.133**



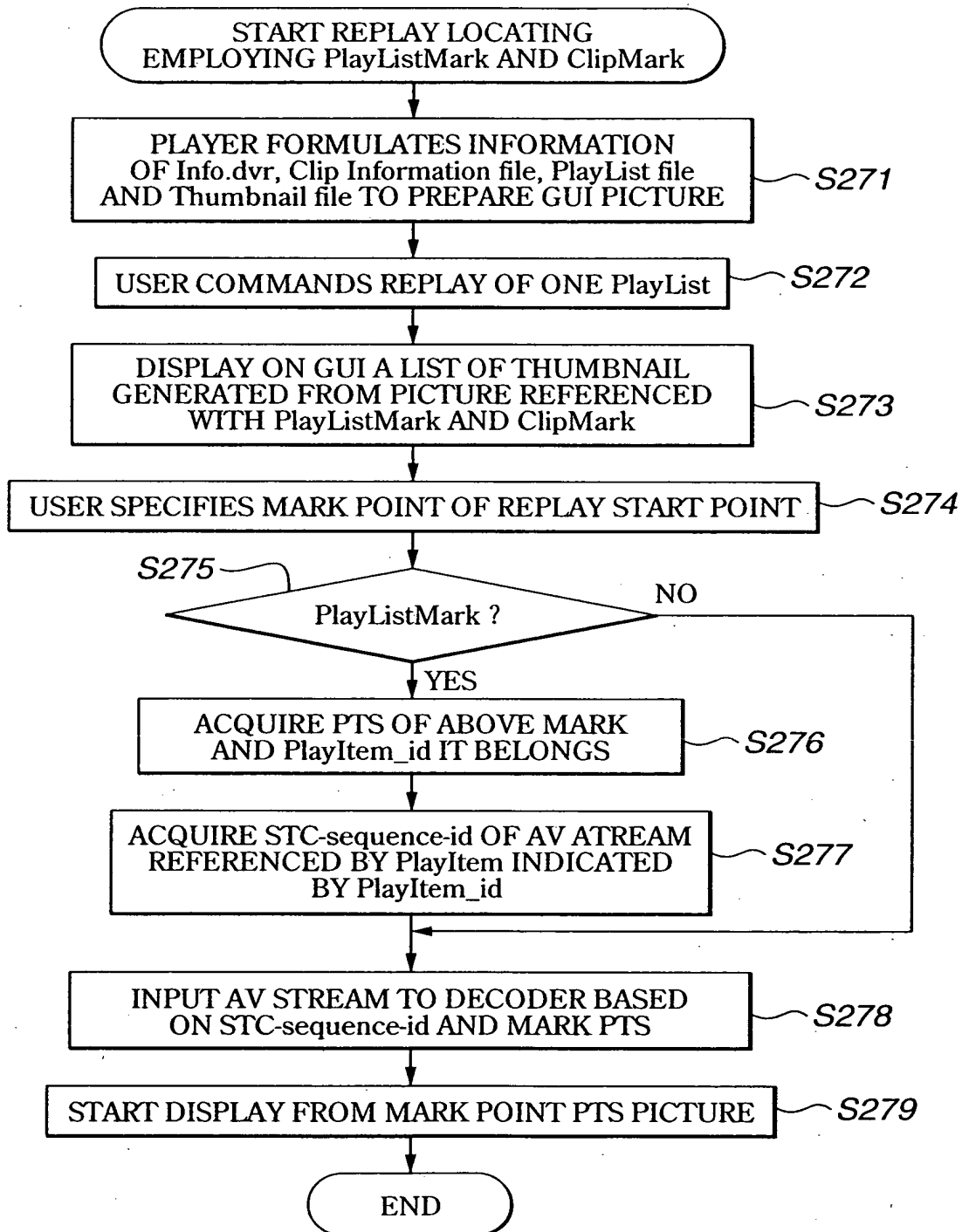


FIG.134

SYNTAX	NUMBER OF BYTES	ABBREVIATION
PlayListMark(){		
length	32	uimsbf
number_of_PlayList_marks	16	uimsbf
for (i=0;i<number_of_PlayList_marks;i++){		
mark_invalid_flag	1	uimsbf
mark_type	7	uimsbf
mark_name_length	8	uimsbf
ref_to_PlayItem_id	16	uimsbf
mark_time_stamp	32	uimsbf
entry_ES_PID	16	uimsbf
ref_to_thumbnail_index	16	uimsbf
mark_name	8*32	bslbf
}		
}		

FIG.135

VALUES	MEANING	NOTS
0x00	Resume-mark	REPLAY RESUME POINT. THE NUMBER OF REPLAY RESUME POINTS DEFINED IN PlayListMark() MUST BE 0 OR 1.
0x01	Book-mark	REPLAY ENTRY POINT OF PlayList. THIS MARK CAN BE SET BY USER AND USED AS MARK FOR SPECIFYING START POINT OF FAVORITE SCENE. A PLURAL NUMBER OF THIS MARK MAY BE PROVIDED IN PlayListMark().
0x02	Chapter-mark	USER INTENDS THAT A CHAPTER IN PlayList BEGINS WITH THIS MARK. A PLURAL NUMBER OF THIS MARK MAY BE PROVIDED IN PlayListMARK().
0x03	Skip-start-mark	IF ONE Skip-start-mark IS SET IN PlayListMark, ONE Skip-end-mark MUST BE SET DIRECTLY AFTER THE Skip-start-mark ENTRY. USER INTENDS TO SKIP PlayList REPLAY FROM TIME STAMP OF Skip-start-mark TO TIME STAMP OF Skip-end-mark. Skip-start-mark HAS SAME MEANING AS Skip-end-mark. IF entry-ES-PID IS NOT 0xFFFF, Skip-start-mark HAS SAME entry-ES-PID VALUE AS Skip-end-mark. THIS MARK CAN BE SET BY USER. THERE MAY BE PLURAL SUCH MARKS IN PlayListMark().
0x04	Skip-end-mark	
0x05- 0x3F	Reserved for future use	Reserved for PlayListMark
0x40- 0x7F	Reserved for ClipMark	

FIG.136

SYNTAX	NUMBER OF BYTES	ABBREVIATION
ClipMark0 {		
<b>length</b>	32	uimsbf
<b>maker_ID</b>	16	uimsbf
<b>number_of_Clip_marks</b>	16	uimsbf
for (i=0; i< <i>number_of_Clip_marks</i> ; i++){		
<b>mark_invalid_flag</b>	1	uimsbf
<b>mark_type</b>	7	uimsbf
<b>ref_to_STC_id</b>	8	uimsbf
<b>mark_time_stamp</b>	32	uimsbf
<b>entry_ES_PID</b>	16	uimsbf
<b>ref_to_thumbnail_index</b>	16	uimsbf
<b>representative_picture_time_stamp</b>	32	uimsbf
}		
}		

FIG.137

Mark_type	MEANING	NOTES
0x00- 0x3F	reserved for future use	Reserved for PlayListMark
0x40	Scene-start-mark	MARK POINT INDICATING SCENE START POINT
0x41- 0x5F	Reserved for common ClipMark	
0x60- 0x7F	Maker defined ClipMark	MAKER SPECIFIED WITH maker_ID CAN FREELY DEFINE MEANING

**FIG.138**

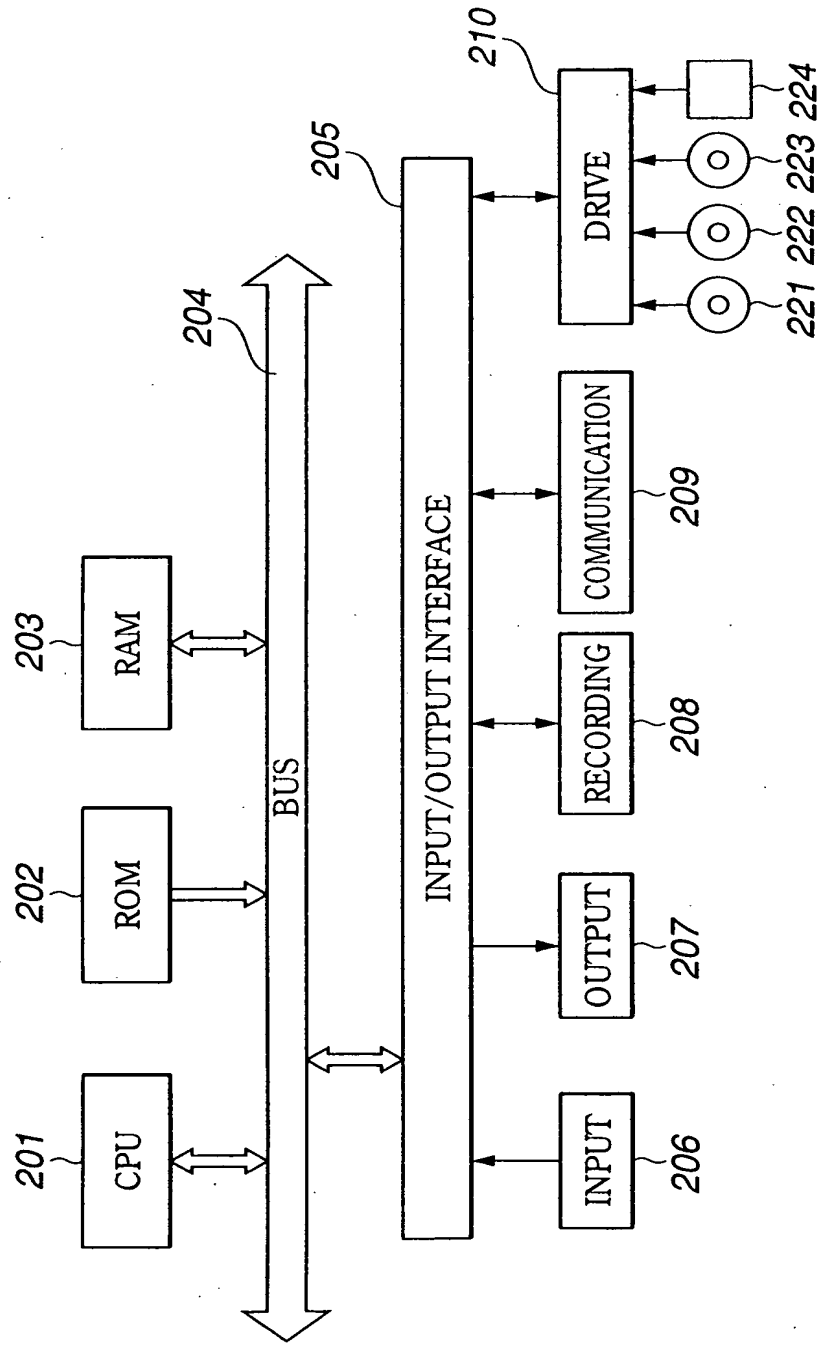


FIG.139